COMPUTING INVARIANTS FOR FINITELY PRESENTED NILPOTENT GROUPS

by

Leon S. Sterling

Thesis presented to the

Australian National University

for the

Degree of Doctor of Philosophy

Canberra

November 1980

Errata for 'Computing Invariants for Finitely Presented Nilpotent Groups'

page 1, 4th line from bottom — delete the first word - presentation .

page 4, 12th line from top — delete the first word - representative .

page 5, 2nd line from top should read
the p-group case holds only for nilpotent groups with torsionfree commutator

page 10, 9th line from top should read
in $I(H)$, then g^n is in H for some non-zero integer n . A normal subgroup H

page 11, 13th line from top should read

(iii) $[a^{-1}, b] = [a, b]^{-1} = [a, b^{-1}]$.

page 15, bottom line
q_{n+} should be replaced by q_n^+

page 19, top line should read

satisfying $u^2 - \Delta v^2 = N$. Solutions are *positive* if $u > 0$ and $v > 0$.

page 20, 4th line should read
PROPOSITION 1.12. *Each class of solutions of* $x^2 - \Delta y^2 = N$ *has a*

page 27, 6th line from bottom should read
A *k*th *determinantal divisor* of a matrix M , denoted $d_k(M)$, is a

page 28, 6th line from top should read
integer matrices, choose the invariant factors to be nonnegative. For $\mathbb{Z}_{(p)}$-

page 30, 7th and 8th lines from bottom should read
example, steps 3 and 4 are done before steps 5 and 6, and step 8 precedes step 9. This relates to the context of presentations of abelian groups

page 37, 14th line from top
3rd column in row G.c.d. should read $65536 = 2^{16}$

page 42, 10th and 11th lines from bottom should read

$$\langle a_1, \ldots, a_d, b_1, \ldots, b_s : [a_j, a_i] = \prod_{k=1}^{s} b_k^{\alpha(i,j,k)}, \ 1 \leq i < j \leq d,$$
$$[b_j, a_i] = \emptyset, \ 1 \leq i \leq d, \ 1 \leq j \leq s,$$

page 47, 4th line from bottom should read

So $M_{\tilde{P}}(i,j) = \sum_{k=1}^{s} \sum_{L=1}^{s} \alpha(i, j, L) S^{-1}(L, k) x_k$.

page 48, 8th line from top should read

$$\beta(i, j, k) = \sum_{L=1}^{s} \sum_{m=1}^{d} \sum_{n=1}^{d} S^{-1}(L, k)\, T(i, m) T(j, n) \alpha(m, n, L) \quad .$$

The IF proof of Theorem 3.3, beginning on the bottom line of page 48 and continuing to the top line of page 50 should be replaced by the following:

Let $\bar{a}_i = \prod_{m=1}^{d} a_m^{T(i,m)}$ and $\bar{b}_k = \prod_{L=1}^{s} b_L^{S^{-1}(k,L)}$.

Let $\bar{P}$ be the canonical presentation for G , relative to the elements $\bar{a}_1, \ldots, \bar{a}_d$ and $\bar{b}_1, \ldots, \bar{b}_s$.

By the above discussion, $M_{\bar{P}} = TM_P{}^S T^t = M_Q$, by assumption.

There is now an obvious isomorphism mapping G to H .

page 51, 4th line from bottom
$[a_r, a_{r-1}]$ should be replaced by $[a_{2r}, a_{2r-1}]$

page 52, 7th line from top should read
the first part of the theorem. Let $r = \lfloor d/2 \rfloor$.

9th line from top and 12th line from top
$[a_r, a_{r-1}]$ should be replaced by $[a_{2r}, a_{2r-1}]$

page 58, 8th line and 3rd line from bottom
$\sum_{k=j}^{d}$ should be replaced by $\sum_{k=j+1}^{d}$

page 64, 2nd line from bottom should read
Else add row 3 to row 1, and go to 9.

page 65, 6th line from bottom should read
$\alpha(1, 4, 2) = q\alpha(1, 3, 2) + r$, $|r| < |\alpha(1, 3, 2)|$.

page 73, 4th line from bottom
The exponent of k_2 should be $\pm((\rho_2+\mu_2)t+(\rho_3+\mu_3)v\lambda)$

page 74, 5th line from bottom
$\gamma_Q\rho_2(\mu_2 u+\mu_3 w)$ should be replaced by $\gamma_Q\rho_4(\mu_2 u+\mu_3 w)$

page 75, 6th line from bottom should read

$$2tuw\lambda - u^2v\lambda^2 = tuw\lambda + u\lambda(tw-uv\lambda) = tuw\lambda \pm u\lambda$$

page 76, 6th line from bottom should read
isomorphism type of the group presented uniquely. The example of

page 84, 8th line from top
$\delta u \pm \delta$ should be replaced by $\gamma u \pm \delta$

page 85, 7th line from bottom
4.14 should be replaced by 4.13

page 91, 7th and 8th lines from top should read

$$v = (-t\delta \pm \sqrt{t^2\delta^2 - 4\gamma\epsilon(t^2 \pm 1)}\ )/2\epsilon$$

$$= (-t\delta \pm \sqrt{t^2\Delta \mp 4\gamma\epsilon}\ )/2\epsilon \quad .$$

page 92, 5th line from top
$\gamma q(p \times \delta q)\epsilon$ should be replaced by $\gamma q(p + \delta q)\epsilon$

9th line from bottom
Δ' should be replaced by Δ

page 94, top line should read
(i) $\Delta < 0$

10th line from top should read

7. If SA_0T^{-1} is in $GL_\lambda(2, \mathbb{Z})$, etc.

page 95, 2nd line from bottom

$SUAT^{-1}$ should be replaced by $SAUT^{-1}$

page 96, 2nd and 3rd lines from top

SUA_0T^{-1} should be replaced by SA_0UT^{-1}

4th and 5th lines from top should read

9. If no $SAUT^{-1}$ is in $GL_\lambda(2, \mathbb{Z})$, where A runs over all elements of Autom(fS), then $f \not\sim_\lambda g$ by Lemma 4.20. //

page 98, 7th line from bottom
$2\gamma(d-a)d$ should be replaced by $2\gamma(d-a)\delta$

page 99, 4th line from bottom
$\begin{bmatrix} \overline{t} & \overline{v} \\ \overline{u} & \overline{w} \end{bmatrix}$ should be replaced by $\begin{bmatrix} \overline{t} & \overline{u} \\ \overline{v} & \overline{w} \end{bmatrix}$

page 100, 4th line from top should read
$v = \overline{t}\gamma q + \overline{v}(p+\delta q)/2$.

On pages 101,102,103,104,105
$\gamma', \delta', \epsilon'$ should be replaced by γ, δ, ϵ respectively.

page 101, 2nd line from top

$SUAT^{-1}$ should be replaced by $SAUT^{-1}$

page 102, 8th line from top

$(p_1^2 - q_1^2)$ should be replaced by $(p_1^2 - \Delta q_1^2)$

10th line from bottom SU should be replaced by S

U^{-1} should be replaced by T^{-1} and U

5th line from bottom

SUA_nT^{-1} should be replaced by SA_nUT^{-1}

page 103, 2nd line from top

$fSUA_iT^{-1}$ should be replaced by fSA_iUT^{-1}

page 104, 3rd line from top should read

$x^2 - \Delta y^2 = 4\gamma\epsilon$.

9th line from top

$\overline{p} = \sqrt{\Delta}\ \overline{q}$ should be replaced by $\overline{p} + \sqrt{\Delta}\ \overline{q}$

page 105, 11th line from bottom

SU and T^{-1} should be replaced by S and UT^{-1} respectively .

10th line from bottom

SUB_nT^{-1} should be replaced by SB_nUT^{-1}

page 110, 6th line from top

$\begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}$ should be replaced by $\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$

page 118, 8th line from bottom

$[a_r, a_{r-1}]$ should be replaced by $[a_{2r}, a_{2r-1}]$

page 121, 9th line from top

$1 \leqslant i \leqslant n$ should be replaced by $1 \leqslant i \leqslant m$

page 125, 4th line from bottom

The 2nd relation should be $[a_3, a_1]^2[a_3, a_2]^{-3} = \emptyset$

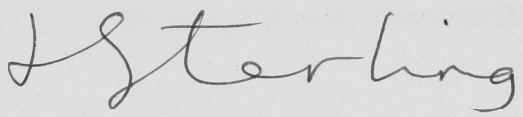
page 165, 7th line from top should read

$I(H)$ (or $I_G(H)$) is the isolator of a normal subgroup H in a group G

STATEMENT

Another account of the material presented in Chapter 2 appeared in a joint paper with Dr G. Havas (Havas and Sterling (1979)).

Unless otherwise stated, the rest of the work presented in this thesis is my own.



Leon Sterling

ACKNOWLEDGEMENTS

The Department of Mathematics in the Institute of Advanced Studies at the Australian National University in Canberra has been an enjoyable place to work. The facilities provided are excellent. The environment is friendly, and the people present in the department, including members of staff, research fellows, visiting fellows, and expecially the secretaries, have been helpful. I thank all concerned.

Three people deserve special mention and thanks for help with the work presented in the thesis. Dr George Havas supervised my work on the computer program related to integer matrices and always had helpful ideas and suggestions. Other people contributing to this part of the work are specifically mentioned in the paper by Havas and Sterling (1979). Dr Fritz Grunewald not only provided hospitality during a visit to Bielefeld in June 1979, but kept up correspondence. He provided preprints of several central papers and made some very influential suggestions on how to present the material. My supervisor, Dr Mike Newman, was always willing to listen to my ideas and made many suggestions throughout the three years. He carefully read drafts of the various chapters and was responsible for many improvements.

For the presentation of the thesis, a very special thanks goes to Mrs B.M. Geary. Her typing is excellent, and her expertise in preparing printed mathematics has been of invaluable assistance. Thanks go also to Ms F.M. Al-Yaman and Ms Z. Sterling for help with proof reading.

Many other people have helped and supported me during the last three years when the work for this thesis has been done. I thank them all collectively.

An Australian National University PhD Scholarship provided financial support during the three years.

ABSTRACT

This thesis is concerned with computing invariants for finitely presented nilpotent groups. Two main problems are considered.

The first discusses practical methods for identifying the isomorphism type of an abelian group from a finite presentation. The classical algorithm computes the Smith normal form of the relation matrix of the presentation. A new algorithm which computes the primary invariants of the associated group rather than the torsion invariants is described. This algorithm appears more effecient for large examples.

The second problem centres on classifying finitely generated, torsion-free nilpotent groups of class 2 . Canonical presentations can be given for each such group. A skew-symmetric matrix can be associated with each canonical presentation. Conditions are given in terms of these matrices for when two canonical presentations present isomorphic groups. An equivalence class of polynomials, related to the determinant of the skew-symmetric matrices, is shown to be an invariant of the group. This is only useful when the matrix has even dimension.

The conditions are used to classify torsionfree nilpotent groups with cyclic commutator subgroup, and groups with Hirsch number less than or equal to 6 . The results for Hirsch number equal to 6 appear new. These groups are determined by the isomorphism type of the commutator quotient group and the equivalence class of a binary quadratic form. A more restricted canonical presentation is given for these groups and a test for determining when two such presentations present isomorphic groups.

Nilpotent groups of class 2 with torsionfree commutator quotient group are also considered. Relational presentations can be given for each such group and a skew-symmetric matrix associated with each relational presentation. Two relational presentations present isomorphic groups when

precisely the same conditions hold for the associated matrices, as for canonical presentations. Thus the classification results hold also for these groups.

Finally, practical algorithms are given for computing both canonical and relational presentations from arbitrary presentations of the respective nilpotent groups of class 2 .

TABLE OF CONTENTS

INTRODUCTION

In his *Lecture Notes on Nilpotent Groups*, Baumslag (1971) wrote:

> *One might hope that for finitely generated nilpotent groups it is possible to effectively obtain a set of reasonable invariants which uniquely determine these groups. Very little information of any kind has been obtained however.*

This thesis describes a set of invariants for some special cases - namely abelian groups, torsionfree nilpotent groups of class 2 , and class 2 groups whose commutator quotient group is a free abelian group.

The computational group theory environment of the Australian National University in Canberra has had a strong influence on this thesis. Both the direction that the work takes and the manner of exposition are affected by computer programs that exist here. In particular, nilpotent groups are considered by way of their finite presentations. Emphasis is placed on computation, and it is demonstrated that the invariants introduced could be practically computed from a large range of finite presentations of nilpotent groups.

The first invariant to be considered arises naturally from the definition of a nilpotent group - namely the class. The theory of finitely generated groups of class 1 , or abelian groups, provides a classification up to isomorphism for these groups. The fundamental theorem of finitely generated abelian groups states that every such group is the direct product of cyclic subgroups, whose orders can be chosen such that a set of invariants for the group is obtained. This set of invariants determines the group up to isomorphism. An integer matrix can be associated with a finite presentation presentation of an abelian group. There is a well-known algorithm for computing a normal form for such a matrix, thereby determining the orders of the subgroups.

This classical algorithm has practical limitations when implemented on

a computer because of the finite resources of the machine. An alternative algorithm has been implemented. This algorithm seems more efficient for identifying finitely presented abelian groups when the numbers of generators and relations in the presentation are sufficiently large. Details of these algorithms are discussed in Chapter 2. A program developed for the identification of abelian groups, and incorporating both algorithms for computing with integer matrices is described. Some large presentations, where the alternative algorithm performed far better are given.

The major part of this work, however, concerns a classification of certain nilpotent groups of class 2 . A finitely generated torsionfree nilpotent group of class 2 can be thought of as representing an alternating bilinear map from $M \times M \to N$, where M and N are free $\mathbb{Z}$-modules. In this way, the classification results can be viewed as a generalisation of the classification of alternating bilinear maps from $V \times V \to W$, where V and W are vector spaces over a field. Metabelian Lie algebras, also known as 2-step nilpotent Lie algebras, can be so considered. Scheuneman (1967) and Gauger (1973) and (1974) discuss these algebras. In several ways, the results presented here can be viewed as a generalisation of results of these papers, pertaining to Lie algebras over the integers. Also related is the work of Brahana (1940) who investigated p-groups of class 2 and exponent p , which are essentially Lie algebras over GF(p) .

In the context of presentations, two desirable objectives of a classification theory of a class of groups are a compactly described list of presentations giving one representative of each isomorphism class of groups, and a recognition algorithm. The latter is a finite procedure to determine which representative on the list presents a group isomorphic to that presented by a given finite presentation of a group in the class. Implicit in this second objective is the isomorphism problem for the class of groups.

Recently, Grunewald and Segal (1979a) solved the isomorphism problem for finitely generated nilpotent groups. Their result leaves the hope that a general classification of these groups using presentations may be possible.

My investigation of nilpotent groups started out ambitiously. The hope was that an existing algorithm could be modified and applied to obtain classification results. The procedure to be adapted was an algorithm for constructing all isomorphism types of finite p-groups described in M.F. Newman (1977). An outline of this algorithm follows.

Consider the series of subgroups of a group G defined by

$$G = P_0(G) \geq P_1(G) \geq \ldots \geq P_i(G) \geq \ldots ,$$

where $P_{i+1}(G) = [P_i(G), G]P_i(G)^p$, $i \geq 0$. $G/P_i(G)$ is a finite p-group for all i . This series terminates in the trivial subgroup iff G is a finite p-group. If $P_c(G)$ is trivial, but $P_{c-1}(G)$ is not, then G has *exponent-p-central class* c .

Let R be a normal subgroup of a free group F of rank d such that F/R is a d-generator finite p-group of exponent-p-central class c . The basic step of the algorithm is to construct a list of isomorphism types of d-generator finite p-groups P of exponent-p-central class $c + 1$, such that $P/P_c(P) \cong F/R$. The list is given in terms of normal subgroups of the free group.

Let G be a finite p-group of exponent-p-central class c , isomorphic to F/R .

The first step of the algorithm is to construct a presentation for the *p-covering group* of G , which is isomorphic to $F/[R, F]R^p$. This can be computed using the nilpotent quotient program described, for example, in Havas and Newman (1980).

Let $R^* = [R, F]R^p$.

The second step calculates a description of $N/R^* = P_c(F/R^*)$ and a description of the set of subgroups U of F such that $R^* \leq U \leq R$ and $UN = R$.

Each automorphism of F/R extends to an automorphism of F/R^* . The isomorphism types of groups sought are in 1-1 correspondence with the orbits of the subgroups mentioned above under the permutations induced from the extended automorphisms just mentioned. This result is the central idea of the algorithm.

The third step calculates an orbit representative, U say, for each of the above orbits. The new group corresponding to U is isomorphic to F/U . The fourth step calculates a stabiliser for each orbit representative representative, allowing the calculation of the automorphism group required for the algorithm to iterate.

In the following, $\Gamma_i(G)$ denotes the ith term of the lower central series of a group G . (Further definitions and notation are given in Chapter 1.) Let G be a nilpotent group of class c . The basic step of the proposed algorithm constructs a list of nilpotent groups H , one of each isomorphism type, such that H is of class $c + 1$ and $H/\Gamma_{c+1}(H) \cong G$. Suppose $G \cong F/R$, and let $R^* = [R, F]$. The procedure depends on the following result. The isomorphism types of groups sought are in 1-1 correspondence with orbits of subgroups U of F such that $R^* \leq U \leq R$ and $UN = R$, where $N/R^* = \Gamma_{c+1}(F/R^*)$. The orbits are those obtained from the permutations induced from automorphisms of F/R^* extended from automorphisms of F/R . A special case of this theorem is proved later as Theorem 5.3. The analogous result for metabelian Lie algebras is proved as Theorem 2.1 of Gauger (1973), and is used in that paper as the basis of the classification.

There are difficulties with applying the modified procedure. The

natural adaptation obtained by modifying the proof of the central result for the p-group case holds only nilpotent groups with torsionfree commutator quotient group. (In keeping with later terminology these will be called *isolated* groups.) Thus initial investigations were restricted to these groups. Another problem, which has not been resolved satisfactorily in general, is choosing the appropriate orbit representative. An anticipated further difficulty is calculating the stabilisers of the orbit representatives.

Nevertheless, the method was applied to some simple cases of nilpotent groups of class 2 . A list of isomorphism types of isolated nilpotent groups on 2 and 3 generators, and of isolated groups on d generators with Hirsch number $d + \binom{d}{2} - 1$ were produced. (These results are given in Chapter 5.) The first case the algorithm could not handle was the family of isolated nilpotent groups of class 2 on 4 generators with Hirsch number 8 .

The set of isomorphism types of finite homomorphic images of a finitely generated nilpotent group is an invariant suggested for considering such groups. It is straightforward to prove that the finite quotients determine uniquely the isomorphism type of a finitely generated abelian group. However, Remeslennikov (1967) gave an example of a pair of nonisomorphic torsionfree nilpotent groups of class 4 which cannot be distinguished by their finite quotients. Baumslag (1974) considered the following pair of nilpotent groups of class 2 ,

$G = \langle a, b; a^{25} = 1, [a, b] = a^5 \rangle$ and $H = \langle c, d; c^{25} = 1, [c, d] = c^{10} \rangle$.

G and H have the same finite images, but they are not isomorphic. Pickel (1971), on the other hand, proved a positive result showing that there are only finitely many isomorphism classes of finitely generated nilpotent groups having the same finite quotients.

Grunewald and Scharlau (1979) joined the various strands. They considered torsionfree nilpotent groups of class 2 , and proved some classification results. All these results correspond (in a sense that will be clarified later) to the classification results obtained using the modified p-group generation algorithm. Further, Grunewald and Scharlau gave examples of 4 generator groups of Hirsch number 6 , which had arbitrarily many different isomorphism classes of groups with the same finite quotients. Again this family of groups corresponds to the difficult family of groups mentioned previously.

The results obtained in this thesis are given in terms of presentations. A *canonical* presentation is introduced for torsionfree nilpotent groups of class 2 . A skew-symmetric matrix whose entries are linear homogeneous polynomials can be associated with a canonical presentation. Conditions are given, in terms of the skew-symmetric matrices, for two canonical presentations of nilpotent groups to present isomorphic groups. A polynomial connected with the determinant of the skew-symmetric matrix can be associated with the presentation. This polynomial leads to a useful invariant for matrices of even order. An analogous polynomial invariant for metabelian Lie algebras was introduced by Scheuneman (1967).

Torsionfree nilpotent groups of Hirsch number less than or equal to 6 are classified, as well as those with cyclic commutator subgroup. For all bar the difficult case which will be denoted $T(4, 2)$ in the notation to be introduced later, the classification has the desirable properties mentioned earlier.

Groups in $T(4, 2)$ are identified by their commutator quotient group and an equivalence class of binary quadratic forms. The equivalence relation is linear substitution by an element of a particular subgroup of finite index in $GL(2, \mathbb{Z})$. A list of presentations for these groups implies

a list of representatives of equivalence classes of binary quadratic forms.

The equivalence relation extensively studied in the literature is linear substitution by an element of $SL(2, \mathbb{Z})$. For this purpose, binary quadratic forms are classified into three types - definite, zero, and indefinite forms. For the first two types, one representative of each equivalence class is easily given. For indefinite forms, however, the classification results are more complicated. Every indefinite form is equivalent to one of a restricted type, and a finite test is described which determines whether two restricted forms are equivalent.

Mrowka (1964) discussed the equivalence relation used here and attempts to list one representative of each equivalence class of definite forms. His results are more complicated than for equivalence of definite forms under $SL(2, \mathbb{Z})$. Therefore, it appears unlikely that a compactly described list could be obtained for groups in $T(4, 2)$. The classification results here take the following form. Every such group is shown to have a special canonical presentation, and a practical test is described which decides when two such presentations present isomorphic groups. This formulation allows questions about groups in $T(4, 2)$ to be answered.

In the case corresponding to groups in $T(4, 2)$, Gauger (1973) found three metabelian Lie algebras over an algebraically closed field with characteristic different from 2. Their quadratic forms are x^2, xy, and 0 respectively. Brahana (1940) found four p-groups in the relevant case with polynomials 0, xy, x^2, and rx^2 for r a non-square in $GF(p)$.

The layout of the thesis is as follows. The first chapter presents some preliminary results and notation. The program for identifying abelian groups is described in the second chapter. The third chapter discusses torsionfree groups, introduces the polynomial invariant and proves the simpler classification results. The fourth chapter is devoted to $T(4, 2)$.

The fifth chapter discusses isolated groups. Special presentations are introduced for these groups and skew-symmetric matrices associated with the presentations. The conditions under which the matrices are associated with isomorphic groups are the same as for torsionfree groups. Thus similar classification results hold for isolated groups and these are restated. The sixth and final chapter shows how the special presentations can be computed from an arbitrary presentation.

CHAPTER 1

PRELIMINARIES

This introductory chapter serves two purposes. It establishes some notation, and states some results used in later chapters. There are three sections:

(i) nilpotent groups;

(ii) presentations, and

(iii) continued fractions and Pellian equations.

I. Nilpotent Groups

The notation used in this thesis for discussing nilpotent groups is introduced here. Most of it is standard. Basic results about nilpotent groups can be found in M. Hall (1959) and P. Hall (1969) and other group theory texts. The terminology used for abelian groups follows Chapter Ten of Hartley and Hawkes (1970). All groups to be considered throughout the thesis are finitely generated.

The *commutator*, $[x, y]$, of two elements x, y is $x^{-1}y^{-1}xy$.

The *lower central series* of a group G is a series of subgroups

$$G = \Gamma_1(G) \geq \Gamma_2(G) \geq \ldots \geq \Gamma_i(G) \geq \ldots$$

defined recursively by

$$\Gamma_{i+1}(G) = [\Gamma_i(G), G] \ , \quad i = 1, 2, \ldots .$$

If one of these subgroups is the trivial subgroup, the group is *nilpotent*. For each i, $\Gamma_i(G)$ is a fully invariant subgroup, with the property that $\Gamma_i(G)/\Gamma_{i+1}(G)$ is abelian. If $\Gamma_{c+1}(G)$ is trivial, but $\Gamma_c(G)$ is non-trivial, then G has *class* c. The class is clearly an invariant of a nilpotent group.

$\Gamma_2(G)$ is the commutator subgroup and is more commonly denoted G'. The commutator subgroup of a nilpotent group is *omissible*, that is if $G = \langle A, G' \rangle$ for some set of elements A, then $G = \langle A \rangle$.

The elements of finite order in a nilpotent group, form a normal subgroup, called the *torsion subgroup*. A group is *torsionfree* if its torsion subgroup is trivial. The *isolator* of a normal subgroup H of a group G, denoted $I_G(H)$ (or just $I(H)$ when the context is clear) is defined so that $I_G(H)/H$ is the torsion subgroup of G/H. Thus if g is in $I(H)$, then g^n is in H for some integer n. A normal subgroup H is *isolated* in G if $I_G(H) = H$.

The theory of abelian groups is well-known. (C_α denotes the cyclic group of order α.)

THEOREM 1.1 (The fundamental theorem of finitely generated abelian groups). *Every finitely generated abelian group is the direct product of cyclic subgroups* $C_{\alpha_1}, \ldots, C_{\alpha_t}$ *and a free abelian group of rank* r. *Further* $\alpha_1 | \alpha_2 | \ldots | \alpha_t$.

The positive integers $\alpha_1, \ldots, \alpha_t$ are the *torsion invariants* of the abelian group, and r is the *torsionfree rank*. The torsion invariants and torsionfree rank uniquely determine the isomorphism type of an abelian group.

Matrices with integers for entries, to be called *integer matrices*, arise in many contexts in the subsequent chapters. Most of the usage centres on $GL(d, \mathbb{Z})$, the group of invertible $d \times d$ integer matrices. Invertible integer matrices are precisely those matrices with determinant ± 1.

The automorphism group of the free abelian group of rank r is $GL(r, \mathbb{Z})$.

THEOREM 1.2. *Let* $\{a_i\}$, $\{b_i\}$ *be two bases for the free abelian group of rank* r . *Then there exists an element* T *of* $\mathrm{GL}(r, \mathbb{Z})$ *such that*

$$a_i = \prod_{j=1}^{r} b_j^{T(i,j)} , \quad 1 \leq i \leq r .$$

The *Hirsch number* of a nilpotent group G , denoted $h(G)$, is the sum of the torsionfree ranks of the abelian sections $\Gamma_i(G)/\Gamma_{i+1}(G)$. The Hirsch number is an invariant of a group.

Most of this thesis is concerned with nilpotent groups of class 2 . For these groups, $G' \leq Z(G)$, the centre of G , and the well-known commutator identities simplify as follows.

LEMMA 1.3. *In a nilpotent group of class* 2 ,

(i) $[a, bc] = [a, b][a, c]$,

(ii) $[ab, c] = [a, c][b, c]$,

(iii) $[a^{-1}, b] = [a, b]^{-1} = [a, b]^{-1}$.

II. Presentations

This description of presentations is modelled on Magnus, Karrass and Solitar (1976).

$\langle a, b, c, \ldots; U(a, b, c, \ldots) = \overline{U}(a, b, c, \ldots),$

$V(a, b, c, \ldots) = \overline{V}(a, b, c, \ldots), W(a, b, c, \ldots) = \overline{W}(a, b, c, \ldots), \ldots \rangle$

is a *presentation*. U, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$, and so on, are words in the symbols $a, b, c, \ldots$. These symbols are called *generators* and the relations $U = \overline{U}$, $V = \overline{V}$, $W = \overline{W}$, ... are called *defining relations*. There is a standard way of associating a unique group with a particular presentation given in section 1.2 of Magnus, Karrass and Solitar (1976).

The presentation is *finitely generated* (*finitely related*) if the number of generators (defining relations) is finite. The presentation is *finite* if

it is both finitely generated and finitely related. The group is then *finitely presented.* This thesis is concerned only with finitely presented nilpotent groups. Discussion in this section is therefore restricted to finite presentations.

One nonstandard notational feature used is $\emptyset$ for the identity of a group. This often makes group presentations clearer to read. A similar usage is found in M.F. Newman (1976a).

A group can have many different presentations. For example,

$$\langle a, b, c; [b, a] = c^2, [c, a] = \emptyset, [c, b] = \emptyset \rangle$$

and

$$\langle x, y, z; [y, x] = z^{-2}, xz = zx, z^2 = x^{-1}zxy^{-1}zy \rangle$$

present isomorphic groups (to be proved later).

The *isomorphism problem* for a class of groups is to determine whether the groups associated with two given presentations of groups in that class are isomorphic. The problem is in general unsolvable. However, Grunewald and Segal (1979a) recently solved the isomorphism problem for finitely presented nilpotent groups. Their methods, as described in Grunewald and Segal (1979b), are involved and not easy to apply to particular presentations. A major part of this thesis describes techniques more amenable for hand or machine calculation to solve the problem in particular cases. These techniques involve the manipulation of presentations.

Certain transformations, called *Tietze transformations*, can be applied to a presentation without changing the isomorphism type of the group being presented. Let the presentation be

$$\langle a_1, \ldots, a_n; W_1 = \overline{W}_1, \ldots, W_m = \overline{W}_m \rangle .$$

The four types of Tietze transformations are:

T1: Add the relation $X = \overline{X}$ if it can be derived from the m defining relations $W_i = \overline{W}_i$, $i = 1, \ldots, m$;

T2: If one of the defining relations $W_i = \overline{W}_i$ can be derived from the others, delete it;

T3: If X is a word in $a_1, \ldots, a_n$, then add x to the set of generators and the relation $x = X$ to the set of defining relations;

T4: If $x = X$ is a relation, where X is a word in the generators other than x, then delete x from the generators, delete $x = X$ from the defining relations and replace x by X in the remaining relations.

The Tietze transformations specified here are more properly called *elementary* Tietze transformations. However, only these transformations need be considered here, as all presentations will be finite.

THEOREM 1.4. *Two finite presentations P, Q present isomorphic groups iff P can be obtained from Q by a finite application of Tietze transformations.*

A proof of this theorem is found on page 52 of Magnus, Karrass and Solitar (1976).

The following example illustrates the use of Tietze transformations to change one presentation into another. Recall the pair of presentations given earlier in this section:

$$\langle a, b, c; [b, a] = c^2, [c, a] = \emptyset, [c, b] = \emptyset \rangle ,$$

and

$$\langle x, y, z; [y, x] = z^{-2}, xz = zx, z^2 = x^{-1}zxy^{-1}zy \rangle .$$

Now,

$$xz = zx \Rightarrow \emptyset = x^{-1}zxz^{-1} \Rightarrow zx^{-1}z^{-1}x = \emptyset$$

or

$$[z^{-1}, x] = \emptyset .$$

Also $xz = zx \Rightarrow z = x^{-1}zx$. Substituting in $z^2 = x^{-1}zxy^{-1}zy$ gives

$$z^2 = zy^{-1}zy \Rightarrow z = y^{-1}zy \Rightarrow zy^{-1}z^{-1}y = \emptyset$$

or

$$[z^{-1}, y] = \emptyset .$$

Using two T1 transformations, the second presentation becomes

$$\langle x, y, z; [y, x] = z^{-2}, xz = zx, z^2 = x^{-1}zxy^{-1}zy,$$
$$[z^{-1}, x] = \emptyset, [z^{-1}, y] = \emptyset \rangle .$$

Reversing the chains of reasoning above shows that $xz = zx$ and $z^2 = x^{-1}zxy^{-1}zy$ are consequences of $[z^{-1}, x] = \emptyset$ and $[z^{-1}, y] = \emptyset$.

Applying two T2 transformations gives

$$\langle x, y, z; [y, x] = z^{-2}, [z^{-1}, x] = \emptyset, [z^{-1}, y] = \emptyset \rangle .$$

Three T3 transformations change the presentation to

$$\langle x, y, z, a, b, c; [y, x] = z^{-2}, [z^{-1}, x] = \emptyset, [z^{-1}, y] = \emptyset,$$
$$a = x, b = y, c = z^{-1} \rangle .$$

Now three T1 transformations are applied to give

$$\langle x, y, z, a, b, c; [y, x] = z^{-2}, [z^{-1}, x] = \emptyset, [z^{-1}, y] = \emptyset,$$
$$a = x, b = y, c = z^{-1}, x = a, y = b, z = c^{-1} \rangle .$$

Three T2 transformations change the presentation to

$$\langle x, y, z, a, b, c; [y, x] = z^{-2}, [z^{-1}, x] = \emptyset, [z^{-1}, y] = \emptyset,$$
$$x = a, y = b, z = c^{-1} \rangle .$$

Finally three T4 transformations leave the presentation as

$$\langle a, b, c; [b, a] = c^2, [c, a] = \emptyset, [c, b] = \emptyset \rangle .$$

An application of Theorem 1.4 shows that the groups associated with the two presentations are isomorphic, as previously claimed.

Theorem 1.4 will be used often, sometimes implicitly. In subsequent applications, however, the sequence of Tietze transformations will not be

exactly specified. The example above, then, illustrates what is involved in a precise specification.

III Continued Fractions and Pellian Equations

The discussion of indefinite binary quadratic forms in Chapter 4 uses results about continued fractions and Pellian equations. These are gathered together in this section, from various texts. The description of infinite continued fractions given here closely follows Chapter IV of Davenport (1970). Chapter X of Hardy and Wright (1960), an alternative treatment, is also referred to. Proofs of all results stated on continued fractions can be found in these two books. Sources for the discussion of solutions to the Diophantine equation $x^2 - \Delta y^2 = N$, for Δ a non-square positive integer, are Nagell (1964) and Leveque (1956). Again, proofs are given in these texts.

Let α be an irrational number. It is shown how to construct the *continued fraction expansion* of α.

Let $q_0 = \lfloor \alpha \rfloor$, the greatest integer less than α. Then $\alpha = q_0 + 1/\alpha_1$, where α_1 is irrational. Then $\alpha_1 = q_1 + 1/\alpha_2$, where $q_1 = \lfloor \alpha_1 \rfloor$. This process can be continued indefinitely. $\alpha_n = q_n + 1/\alpha_{n+1}$, where $q_n = \lfloor \alpha_n \rfloor$, and so on. Thus

$$\alpha = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \ddots \cfrac{1}{q_n + \cfrac{1}{\alpha_{n+1}}}}}.$$

This can be written more compactly as

$$\alpha = q_0 + \frac{1}{q_1+}\,\frac{1}{q_2+} \cdots \frac{1}{q_{n+}}\,\frac{1}{\alpha_{n+1}}.$$

Note that $\alpha_i > 1$ for all i, and hence q_i is a positive integer for $i \geq 1$.

The numbers $q_0, q_1, \ldots$ are called the *terms* or the *partial quotients* of the continued fraction, and the *complete quotient* corresponding to q_n is $\alpha_n = q_n + 1/\alpha_{n+1}$, that is α_n is the nth complete quotient.

The *convergents* to the continued fraction are

$$A_0/B_0 = q_0,\ A_1/B_1 = q_0 + 1/q_1 = (q_0q_1+1)/q_1,$$

$$A_2/B_2 = q_0 + 1/(q_1+1/q_2) = (q_0q_1q_2+q_0+q_2)/(q_1q_2+1), \ldots,$$

A_n/B_n being the nth convergent.

Expressions for the A_i's and B_i's can be given by a bracket formula, defined recursively.

Let $[q_0] = q_0$, $[q_0, q_1] = q_0q_1 + 1$, and

$$[q_0, q_1, \ldots, q_n] = q_0[q_1, \ldots, q_n] + [q_2, \ldots, q_n].$$

Then

$$A_i/B_i = [q_0, \ldots, q_i]/[q_1, \ldots, q_i].$$

One can use this recurrence relation to prove

LEMMA 1.5. *If A/B is a convergent in the continued fraction expansion of α, then B/A is a convergent in the expansion of $1/\alpha$.*

The recurrence relation can also be written as

$$[q_0, q_1, \ldots, q_n] = q_0[q_0, q_1, \ldots, q_{n-1}] + [q_0, q_1, \ldots, q_{n-2}].$$

Successive convergents have many properties. Useful ones are

$A_mB_{m-1} - B_mA_{m-1} = (-1)^{m-1}$, and

LEMMA 1.6. $\alpha = (A_n\alpha_{n+1}+A_{n-1})/(B_n\alpha_{n+1}+B_{n-1})$.

Convergents also give good approximations to the irrational. It is true that $|\alpha-(A_n/B_n)| < 1/B_n^2$. The converse is true for a sharper

inequality.

LEMMA 1.7. *If* $|\alpha-(A/B)| < 1/2B^2$, *then* A/B *is a convergent in the continued fraction expansion of* α .

Suppose two complete quotients are equal, that is $\alpha_m = \alpha_n$ for $n > m$. Then the terms $q_{n+1}, \cdots, q_{2n-m}$ will be the same as the terms $q_{m+1}, \cdots, q_n$ and these $n - m$ partial quotients will recur. These terms are called the *period.* In this case the continued fraction is *periodic*, with period of *length* $n - m$, starting from the mth term. A periodic continued fraction is denoted $q_0, q_1, \cdots, q_m, \overline{q_{m+1}, \cdots, q_n}$ where the elements under the bar are the period.

LEMMA 1.8. *Irrational roots of quadratic equations with integer coefficients have periodic continued fractions.*

Of particular interest is the continued fraction of $\sqrt{\Delta}$, where Δ is a non-square positive integer.

LEMMA 1.9. *The continued fraction for* $\sqrt{\Delta}$ *is necessarily of the form* $q_0, \overline{q_1, q_2, \cdots, q_2, q_1, 2q_0}$.

The continued fraction for $\sqrt{\Delta}$ can be used to obtain a solution to the Diophantine equation $x^2 - \Delta y^2 = 1$, known as *Pell's equation.*

Let $\sqrt{\Delta}$ have the continued fraction given in Lemma 1.9, and let n be the length of the period. Then A_{n-2}/B_{n-2} and A_{n-1}/B_{n-1} are the two convergents coming immediately before the $2q_0$ term.

By Lemma 1.6,

$$\sqrt{\Delta} = \left(\alpha_n A_{n-1}+A_{n-2}\right)/\left(\alpha_n B_{n-1}+B_{n-2}\right) . \tag{1}$$

Now

$$\alpha_n = 2q_0 + \frac{1}{q_1+} \cdots = \sqrt{\Delta} + q_0 .$$

Substituting in (1) and multiplying through, gives

$$\sqrt{\Delta}\left(\sqrt{\Delta}+q_0\right)B_{n-1} + \sqrt{\Delta}B_{n-2} = \left(\sqrt{\Delta}+q_0\right)A_{n-1} + A_{n-2} .$$

Equating the coefficients of $\sqrt{\Delta}$ leads to the two equations

$$q_0B_{n-1} + B_{n-2} = A_{n-1} \tag{2}$$

and

$$\Delta B_{n-1} = q_0A_{n-1} + A_{n-2} . \tag{3}$$

Recall that $A_{n-1}B_{n-2} - B_{n-1}A_{n-2} = (-1)^{n-2}$. Substituting for A_{n-2} and B_{n-2} using equations (2) and (3) gives

$$A_{n-1}\left(A_{n-1}-q_0B_{n-1}\right) - B_{n-1}\left(\Delta B_{n-1}-q_0A_{n-1}\right) = (-1)^{n-2}$$

or

$$A_{n-1}^2 - \Delta B_{n-1}^2 = (-1)^{n-2} .$$

If n is even, then A_{n-1}, B_{n-1} satisfy Pell's equation. If n is odd, consider $\alpha_{2n} = \alpha_n = \sqrt{\Delta} + q_0$.

Substituting in

$$\sqrt{\Delta} = \left(\alpha_{2n}A_{2n-1}+A_{2n-2}\right)/\left(\alpha_{2n}B_{2n-1}+B_{2n-2}\right) ,$$

which holds from Lemma 1.6, and continuing as above, gives

$$A_{2n-1}^2 - \Delta B_{2n-1}^2 = (-1)^{2n-2} = 1 .$$

This method is demonstrated on pages 108 and 109 of Davenport (1970) to find a solution to $x^2 - 21y^2 = 1$ and $x^2 - 29y^2 = 1$. Note that it proves that Pell's equation always has a nontrivial solution.

Pell's equation is a particular example of the Diophantine *Pellian equation* $x^2 - \Delta y^2 = N$. The rest of this section discusses this general equation. Throughout, Δ is a non-square positive integer. When results/notation hold specifically for $N = 1$, Pell's equation will still be referred to. Some definitions and notation are introduced.

$u + v\sqrt{\Delta}$ is a *solution* of $x^2 - \Delta y^2 = N$, if u and v are integers

satisfying $u^2 = \Delta v^2 = N$. Solutions are *positive* if $u > 0$ and $v > 0$.

Positive solutions can be ordered by the size of u. The minimal positive solution of Pell's equation is called the *fundamental solution*. It can be proved that the continued fraction method described above for solving Pell's equation yields the fundamental solution.

PROPOSITION 1.10. *Suppose that* $u + v\sqrt{\Delta}$ *is a solution of the Pellian equation* $x^2 - \Delta y^2 = N$, *and* $x + y\sqrt{\Delta}$ *is a solution of Pell's equation. Then* $(u+v\sqrt{\Delta})(x+y\sqrt{\Delta})$ *is a solution of the Pellian equation.*

Proof. $(u+v\sqrt{\Delta})(x+y\sqrt{\Delta}) = (ux+\Delta yv) + (vx+uy)\sqrt{\Delta}$.

Now

$$(ux+\Delta yv)^2 - \Delta(vx+uy)^2 = u^2x^2 + 2\Delta uxyv + \Delta^2y^2v^2 - \Delta v^2x^2 - 2\Delta vxuy - \Delta u^2y^2$$

$$= x^2\left(u^2-\Delta v^2\right) - \Delta y^2\left(u^2-\Delta v^2\right)$$

$$= \left(u^2-\Delta v^2\right)\left(x^2-\Delta y^2\right) = N\cdot 1 = N . \qquad \square$$

This proposition establishes the result that there are infinitely many solutions to Pell's equation, provided by the powers of the fundamental solution.

PROPOSITION 1.11. *All solutions of Pell's equation are given by* $x + y\sqrt{\Delta} = \pm\left(x_1+y_1\sqrt{\Delta}\right)^n$, *where* $x_1 + y_1\sqrt{\Delta}$ *is the fundamental solution, and* n *is any integer.*

A description of the solutions, if they exist, of the Pellian equation is more complicated.

Two solutions $u + v\sqrt{\Delta}$ and $u' + v'\sqrt{\Delta}$ are *associated* if $u + v\sqrt{\Delta} = (u'+v'\sqrt{\Delta})(x+y\sqrt{\Delta})$ for some solution $x + y\sqrt{\Delta}$ of Pell's equation. The set of all solutions associated with each other is called a *class* of solutions of a Pellian equation. Clearly, there are infinitely many solutions in a class, given by $u + v\sqrt{\Delta} = (u'+v'\sqrt{\Delta})(x+y\sqrt{\Delta})^n$. A necessary and sufficient condition for two solutions $u + v\sqrt{\Delta}$ and $u' + v'\sqrt{\Delta}$ of

$x^2 - \Delta y^2 = N$ to be associated is that the numbers $uu' - vv'\Delta$ and $vu' - uv'$ are divisible by N. This information can be used to obtain a finite test for the solvability of a Pellian equation.

PROPOSITION 1.12. *Each class of solutions of* $x^2 - \Delta y^2 = N$ *has a solution* $u + v\sqrt{\Delta}$ *with*

(i) $0 < u \leq \sqrt{\tfrac{1}{2}(x_1+1)N}$, *if* $N > 0$,

(ii) $0 \leq u \leq \sqrt{\tfrac{1}{2}(x_1-1)\cdot|N|}$, *if* $N < 0$,

where $x_1 + \sqrt{\Delta}y_1$ *is the fundamental solution of* $x^2 - \Delta y^2 = 1$.

This proposition yields the result that there are only a finite number of classes of solutions of a given Pellian equation. Representatives of the classes can be found by testing whether $(u^2-N)/\Delta$ is a square for the values of u in the interval given in Proposition 1.12. If there are several solutions $u + v\sqrt{\Delta}$ arising, it is easy to decide which are in the same class.

Two particular Pellian equations specified in Chapter 4 are $x^2 - \Delta y^2 = 4$ and $x^2 - \Delta y^2 = -4$. A complete list of the solutions in these cases can be given.

PROPOSITION 1.13. *If* $p_1 + q_1\sqrt{\Delta}$ *is the minimal positive solution of the equation* $x^2 - \Delta y^2 = 4$, *then all solutions are given by*

$$p + q\sqrt{\Delta} = \pm 2\left((p_1+q_1\sqrt{\Delta})/2\right)^n, \quad n \in \mathbb{Z}.$$

The equation $x^2 - \Delta y^2 = 4$ is always solvable, since $2x_1 + 2y_1\sqrt{\Delta}$, where $x_1 + y_1\sqrt{\Delta}$ is the fundamental solution of Pell's equation, is a solution.

PROPOSITION 1.14. *If the equation* $x^2 - \Delta y^2 = -4$ *is solvable and*

$p_1 + q_1\sqrt{\Delta}$ *is its minimal positive solution, then all solutions are given by*

$$p + q\sqrt{\Delta} = \pm 2\left((p_1+q_1\sqrt{\Delta})/2\right)^{2n+1} , \quad n \in \mathbb{Z} .$$

CHAPTER 2

A PROGRAM FOR IDENTIFYING ABELIAN GROUPS

This chapter presents a computer program for computing normal forms of integer matrices. A listing of the code, together with a primitive user's guide, appears in the appendix. The program has been used to identify certain abelian groups.

An integer matrix, known as a *relation matrix*, can be associated with a finite presentation of an abelian group. A good account of the relationship between integer matrices and abelian groups can be found in Chapter Ten of Hartley and Hawkes (1970). The isomorphism type of the abelian group is uniquely determined by the *Smith normal form* of one of its relation matrices.

There is an algorithm for computing the Smith normal form of an integer matrix which is well known. A version is described in most standard algebra textbooks. Throughout this chapter it will be called the *basic algorithm*.

The basic algorithm, as usually described, has a serious deficiency in practical applications. During intermediate stages of calculation, entries in the matrix become inordinately large. This will be called *entry explosion*. (It is sometimes referred to in the literature as *intermediate expression swell* - see, for example, Cabay and Lam (1979).) Computers have limits on the size of integers that can be easily stored and used in calculations. When the size of entries exceeds this limit, *integer overflow* occurs. Thus, if entry explosion leads to integer overflow during computations using the basic algorithm, different techniques will have to be adopted in order to compute the Smith normal form.

An initial motivation for the program presented here was the desire to find the invariant factors of a particular 304×153 relation matrix. The

only non-zero entries were 1's and -1's sparsely spread throughout the matrix. An implementation of the basic algorithm was unsuccessful because of integer overflow. Details about the background of this example and a history of computations are given in Section IV.

The successful approach, which determined the invariant factors of this matrix, involved computing directly the primary invariants of the associated group rather than the torsion invariants. This also led to a new algorithm for computing the Smith normal form of an integer matrix. Gerstein (1977) independently described this alternative approach.

The program consists of implementations of the new algorithm and the basic algorithm, and various heuristic modifications made to improve the performance of the basic algorithm. These have been refined during subsequent applications. It is hard to quantify in general terms the performance of the various components of the program. The examples of Section IV illustrate this.

It seems difficult, also, to establish claims about the program's relative performance. Several computer implementations of the basic algorithm have been outlined. D.A. Smith (1966), Bradley (1971), Kannan and Bachem (1979) among others, all describe programs handling integer matrices. In discussing their variants, however, most authors concentrate on theoretical performance. Practical details of implementations, or statistics about the range of matrices their implementation can successfully handle, are generally omitted. There are, indeed, many interesting unsolved questions about algorithms for computing the Smith normal form. They belong to the realm of complexity theory, however, and will not be discussed here. The emphasis in this description is on practical performance.

An important influence on the program has been the Reidemeister-Schreier algorithm. This algorithm computes a presentation for a subgroup of finite index of a given group. The presentations produced by the

algorithm are in general unwieldy, have many generators and relations, and it is difficult to extract information about the subgroup. By abelianizing the relations, a presentation for the commutator quotient group is obtained. The isomorphism type of this maximal abelian quotient group is often a very useful piece of information. This is determined by the Smith normal form of the relation matrix. Almost all applications of the program have been in this context. The particular implementation of the Reidemeister-Schreier algorithm used is described in Havas (1974). Various features of the program described here were determined by the nature of the presentations produced by the Reidemeister-Schreier program.

Although the main application has been to compute the Smith normal form, the program can be adapted for other, related calculations. One series of calculations is worth mentioning. Presentations of subgroups of

$$\langle a, b, c; b^2 = (ab)^3 = [a, c] = (acbc^{-2}b)^2 = (acbc^{-1}b)^3 = a^2cbc^{-1}(bc)^2bc^{-1}b = \emptyset \rangle$$

were obtained using the Reidemeister-Schreier program. The rank modulo a small prime (< 200) of the relation matrix of the commutator quotient group of the subgroup was computed. For a subgroup of index 540, this involved computing with a 4311×1081 matrix. The program was used with only minor modifications. However, the range of applicability of the program remains largely unexplored.

This chapter, therefore, discusses the program in the context of computing the Smith normal form of an integer matrix. The first section establishes the theory of the Smith normal form. The version of the basic algorithm used by the program is given in the second section, together with the new algorithm. The third section describes some of the techniques used to implement the new algorithm, while the final section gives some applications of the program. The program's performance for these examples is some indication of the range of the various methods.

I. Smith Normal Form

In this chapter, the underlying problem is to compute a normal form for $m \times n$ integer matrices under equivalence. (Equivalence is a term used also in Chapter 4 in connection with binary quadratic forms. This duplication of terminology is unfortunate, but both usages of the word 'equivalence' are firmly established in the literature. The context will, it is hoped, make sufficiently clear which meaning of equivalence is being used.) Two $m \times n$ integer matrices, A and B, are *equivalent* if there is an element P of $\mathrm{GL}(m, \mathbb{Z})$ and an element Q of $\mathrm{GL}(n, \mathbb{Z})$ such that $B = PAQ$.

The methods implemented in the program involve considering matrices with entries from rings other than the integers. Thus, it is convenient to consider a slightly more general problem for the purposes of this chapter - namely equivalence of matrices whose entries come from a Euclidean domain.

DEFINITION 2.1. A *Euclidean domain* R is a commutative ring with identity having no zero divisors, with

1. a function $\emptyset$ from the non-zero elements of R to the non-negative integers satisfying $\emptyset(ab) \geq \emptyset(a)$,
2. an algorithm which when given two elements a, b of R, with $b \neq 0$, computes q, r such that $a = qb + r$ with $r = 0$ or $\emptyset(r) < \emptyset(b)$.

For example, $\mathbb{Z}$, the ring of integers is a Euclidean domain with $\emptyset(a) = |a|$. The division algorithm for two integers satisfies the second condition. In general, the algorithm associated with a Euclidean domain *via* requirement 2 above will be called a *division algorithm*.

The above definition is different from the standard definition of a Euclidean domain. This is because it insists on there being an algorithm to calculate the elements q, r - rather than just guaranteeing the existence of such elements. Clearly all Euclidean domains are Euclidean in the

standard sense, but the reverse implication is by no means clear. However, every known Euclidean domain does have an algorithm. The only discussion relevant to this point that I am aware of in the literature is the work of Samuel (1971).

Consider the localisation of the integers at a prime p, to be denoted $\mathbb{Z}_{(p)}$. $\mathbb{Z}_{(p)}$ is the ring of fractions of the form m/n, $m, n \in \mathbb{Z}$ and $p \nmid n$. Every non-zero element in $\mathbb{Z}_{(p)}$ can be written in the form $p^{\alpha}.k/n$, where $p \nmid k$, n, and $\alpha \geq 0$. Further, all such expressions occur. Note that in this case k/n is a unit, since $(k/n)(n/k) = 1$. Thus every non-zero element of $\mathbb{Z}_{(p)}$ has the form up^{α} for some unit u. Further if $s = up^{\alpha}$ and $t = vp^{\beta}$, then $s|t$ iff $\alpha \leq \beta$.

Consider the map from the non-zero elements of $\mathbb{Z}_{(p)}$ to the non-negative integers, given by $\emptyset(up^{\alpha}) = \alpha$. Let $s = up^{\alpha}$ and $t = vp^{\beta}$. Then $\emptyset(st) = \emptyset(uvp^{\alpha+\beta}) = \alpha + \beta \geq \alpha = \emptyset(s)$.

A division algorithm for $\mathbb{Z}_{(p)}$ is as follows. Let s, t be two elements of $\mathbb{Z}_{(p)}$ with $t \neq 0$. Let $t = vp^{\beta}$.

1. If $s = 0$, then $s = 0.t + 0$.

Let $s = up^{\alpha}$.

2. If $\alpha < \beta$, then $s = 0.t + s$.

3. If $\alpha \geq \beta$, then $s = (u/v)p^{\alpha-\beta}.t + 0$.

Thus $\mathbb{Z}_{(p)}$ is a Euclidean domain.

For the rest of this section R denotes a Euclidean domain. A matrix with entries from R is called an *R-matrix*. Proofs of all claims about R-matrices can be found in Chapter 1 of Morris Newman (1972). $GL(d, R)$ is the group of invertible $d \times d$ R-matrices. If A, B are two $m \times n$ R-matrices, then A is *equivalent* to B if there is an element P of

$GL(m, R)$ and an element Q of $GL(n, R)$ such that $B = PAQ$.

The *elementary row (column) operations* on an R-matrix are:

1. multiply a row (column) by a unit: (for the integers, the units are ± 1),
2. interchange two rows (columns),
3. add a multiple of one row (column) to another.

To each of these elementary row (column) operations there corresponds a unique R-matrix. Multiplying on the left (right) by that matrix has the same effect as performing the elementary row (column) operation. These matrices are called *elementary matrices.*

LEMMA 2.2. *Every invertible square R-matrix is the product of finitely many elementary matrices.*

This is proved on page 24 of Morris Newman (1972). Alternatively a proof can be distilled from the algorithm of the next section. Thus the problem of determining a normal form for R-matrices under equivalence is precisely the problem of computing a normal form for R-matrices under the application of finitely many elementary row and column operations.

The problem of determining a normal form for integer matrices under equivalence was solved by H.J.S. Smith (1861). For general R-matrices, the solution appears to have been well-known and appeared in the literature as soon as the language of Euclidean rings evolved. (See for example Macduffee (1933).) A matrix D is *diagonal* if $D(i, j) = 0$ for $i \neq j$. The kth *determinantal divisor* of a matrix M, denoted $d_k(M)$ is a greatest common divisor of the determinants of all $k \times k$ submatrices of M. Determinantal divisors have the property that $d_{k-1}(M) | d_k(M)$. For convenience, $d_0(M)$ is defined to be 1. A kth *invariant factor* is $d_k(M)/d_{k-1}(M)$, and is denoted $e_k(M)$. Again $e_0(M)$ is defined to be 1. Invariant factors are determined up to multiplication by a unit, and have

the property that $e_{k-1}(M) | e_k(M)$.

THEOREM 2.3. *Every matrix M with entries from a Euclidean domain is equivalent to a diagonal matrix whose entries are invariant factors of M .*

For the cases to be considered, namely integer matrices and $\mathbb{Z}_{(p)}$-matrices, a particular choice of the invariant factors is made. For integer matrices, choose the invariant factors to be positive. For $\mathbb{Z}_{(p)}$-matrices, choose the invariant factors to have the form 0 or p^{α} . The diagonal matrix of Theorem 2.3 is then uniquely determined, and is known as the *Smith normal form* of the given matrix M . It is denoted $S(M)$ and $S_p(M)$ respectively for $\mathbb{Z}$- and $\mathbb{Z}_{(p)}$-matrices. Equivalent matrices have the same invariant factors.

Consider the relation matrix of a finite presentation of an abelian group. (Relation matrices are defined in Section I of Chapter 6.) Note that integer matrices are also $\mathbb{Z}_{(p)}$-matrices. The nontrivial (not equal to 0 or 1) invariant factors of the relation matrix are the torsion invariants of the abelian group. The nontrivial invariant factors of the relation matrix, considered as a $\mathbb{Z}_{(p)}$-matrix, are the p-primary invariants of the group. The torsionfree rank of the group is the difference between the number of columns of the matrix and its rank.

The rest of this chapter presents methods to compute the Smith normal form in the context of abelian groups.

II. A Diagonalisation Algorithm

This section is introduced with the central algorithm of the program. It is along the lines of the more careful textbook descriptions of algorithms for computing Smith normal forms, see for example Section 7.5 of Hartley and Hawkes (1970). The procedure here is for matrices with entries from a

Euclidean domain, and is implemented in the program for both integer matrices (subroutine IMDIAG) and $\mathbb{Z}_{(p)}$-matrices (subroutine MODIAG). Relevant features of the algorithm and its implementations will be indicated shortly.

Let M be a matrix with entries from a Euclidean domain, whose function is $\emptyset$ as in Definition 2.1. Then the steps are as follows.

1. If the matrix is the zero matrix, stop.
   Note that the zero matrix is in Smith normal form, considered both as an integer matrix and as a $\mathbb{Z}_{(p)}$-matrix.
2. Find a non-zero matrix element, $M(i, k)$ say, such that $\emptyset(M(i, k))$ is minimal.
3. If $M(i, k)$ divides all the entries in its column, go to 5.
4. Choose $M(j, k)$ not divisible by $M(i, k)$. Use the division algorithm to find q, r such that $M(j, k) = qM(i, k) + r$ where $\emptyset(r) < \emptyset(M(i, k))$. Replace row j by row j minus q times row i. Go to 2.
5. If $M(i, k)$ divides all the entries in its row, go to 7.
6. Choose $M(i, h)$ not divisible by $M(i, k)$. Determine q, r such that $M(i, h) = qM(i, k) + r$ where $\emptyset(r) < \emptyset(M(i, k))$. Replace column h by column h minus q times column k. Go to 2.
7. Shift $M(i, k)$ to the top left hand corner of the matrix by using the appropriate row and column interchanges.
8. Subtract the appropriate multiple of the top row from each other row to make the entries in the leftmost column below the top zero.
9. Make all entries in the top row, except the leftmost entry,

zero by subtracting the appropriate multiple of the leftmost column from each other column. Note that only the top row is affected by this step and is therefore, in practice, omitted.

10. Consider the smaller matrix obtained by deleting the top row and the leftmost column from the current matrix, and go to 1.

The procedure described is an algorithm because the minimal value of $\emptyset(M(i, k))$, as specified in step 2, strictly decreases on each pass through the loops contained in steps 2-6. Thus the loops are traversed only a finite number of times.

For $\mathbb{Z}_{(p)}$-matrices, the implementation can be simplified. The minimal element found in step 2 necessarily divides all other entries in the matrix. Thus steps 3-6 are omitted in practice. Also, step 7 incorporates a normalisation step to facilitate step 8.

For integer matrices, the algorithm computes a diagonal matrix equivalent to the initial matrix rather than its Smith normal form. The invariant factors can then be routinely found by appropriate calculations of greatest common divisors and least common multiples of the diagonal entries. This is implemented in subroutine SMITH. In all our practical applications, however, the invariant factors have been immediately obvious from the diagonal matrix.

Row operations are done in preference to column operations. For example, steps 3 and 4 are done before steps 5 and 6, and step 9 precedes step 8. This relates to the context of presentations of abelian groups and retention, where possible, of the original group generating set.

As mentioned in the introduction to this chapter, entry explosion and subsequent integer overflow are serious problems often preventing successful termination of the basic algorithm. Various heuristic methods have been developed to mitigate the effect of the problem for integer matrices. They

essentially delay the occurrence of entry explosion. These heuristics will not be discussed here. A description of them can be found in Havas and Sterling (1979). Details also appear in the appendix - in both the user's guide and the code of subroutines IMDIAG and REDROW.

The invariant factors of $\mathbb{Z}_{(p)}$-matrices can be calculated by congruential techniques. Suppose that the largest non-zero invariant factor is p^{α}. Then performing computations modulo $p^{\alpha+1}$ will give the correct invariant factors. This is implemented in subroutine MODIAG. This subroutine can also be regarded as computing a normal form for matrices with entries from the ring $\mathbb{Z}/p^{\alpha+1}\mathbb{Z}$. Fuller (1955) describes an algorithm for computing a Hermite canonical form for matrices with elements from this ring. Note that $\mathbb{Z}/p^{\alpha+1}\mathbb{Z} \cong \mathbb{Z}_{(p)}/p^{\alpha+1}\mathbb{Z}_{(p)}$.

The fact that invariant factors of $\mathbb{Z}_{(p)}$-matrices can be computed by modular techniques suggests an alternative approach for computing the Smith normal form of an integer matrix. This approach has also been described by Gerstein (1977), though his motivation for proposing an alternative algorithm was different.

Throughout the rest of the chapter M denotes an $m \times n$ integer matrix of rank r. Note that r is less than or equal to the minimum of m and n. By Theorem 2.3 there are matrices P, Q such that PMQ is diagonal, and the kth diagonal entry is $e_k(M)$. Now $e_k(M) = u_k p^{\alpha(k)}$, where $p \nmid u_k$, $1 \leq k \leq r$. Note that the value of u_k depends on the prime. Further u_k is a unit in $\mathbb{Z}_{(p)}$. Multiplying the kth row by u_k^{-1} for $1 \leq k \leq r$, gives a diagonal matrix $\overline{M}$ such that $\overline{M}(i, i) = p^{\alpha(i)}$ and $0 \leq \alpha(1) \leq \alpha(2) \leq \ldots \leq \alpha(r)$. Thus $\overline{M} = S_p(M)$ by the uniqueness of the Smith normal form. Note that if $p \mid e_k(M)$ then the kth invariant

factor of $S_p(M)$ is 1 .

Conversely the invariant factors of M can be calculated from the diagonal entries of $S_p(M)$ for varying primes p . Let $p_1, \ldots, p_t$ be the prime divisors of $e_k(M)$. Then $e_k(M)$ equals the product of the kth invariant factors of $S_{p_i}(M)$, $1 \leq i \leq t$. Because of the divisibility of invariant factors, it is enough to consider the prime divisors of $e_r(M)$, the largest non-zero invariant factor. Note that $e_r(M) | d_r(M)$ and the rth determinantal divisor can be considered instead. The alternative procedure for computing the Smith normal form of an integer matrix M is outlined as follows.

1. Determine the rank r of the matrix.
2. Calculate a multiple M of the rth determinantal divisor, by computing determinants of $r \times r$ submatrices.
3. Factorize $M = \prod_{p|M} p^{\gamma(p)}$.
4. For each prime p in the factorization of M , compute $S_p(M)$.
5. Multiply the diagonal entries of the various $S_p(M)$'s to give the invariant factors of M .

Steps 4-5 have been discussed in this section. One comment should be made about computing $S_p(M)$. As mentioned, the computations can be done modulo p^{β} , where β is greater than the largest exponent of p in $S_p(M)$. From step 3, β can be chosen to be $\gamma(p) + 1$. If this value is too large to prevent integer overflow, a smaller value of β is chosen. In practice, finding a suitable value has never been a problem.

The next section describes techniques for performing steps 1 to 3.

III Implementation Features

Classical methods for finding the rank of a matrix or calculating determinants depend on elimination techniques, which suffer from entry explosion. Superficially, therefore, the alternative algorithm may seem no better. However congruential techniques can be applied which avoid the problem of entry explosion. Borosh and Fraenkel (1966) and Cabay and Lam (1977) use these techniques in the related context of the exact solution of linear equations. Congruential methods cannot be directly applied to compute the Smith normal form.

The rank of the matrix is provisionally calculated by computing the rank modulo a large prime. This value is correct unless the prime divides the largest non-zero invariant factor, and the likelihood of this happening is very low. Before describing how the rank may be determined exactly, the following lemma is given.

LEMMA 2.4. *If* A *is an* $n \times n$ *integer matrix, then*

$$\det A \leq \prod_{i=1}^{n} \left(\sum_{j=1}^{n} A(i, j)^2 \right)^{\frac{1}{2}} .$$

This well-known inequality is due to Hadamard. A proof appears in Mehta (1977).

Recall that M is an $m \times n$ matrix of rank r. By the above lemma,

$$d_r(M) \leq \prod_{i=1}^{m} \left(\sum_{j=1}^{n} M(i, j)^2 \right)^{\frac{1}{2}} .$$

This bound will be called the Hadamard bound. Choose a number of primes whose product exceeds this bound. Compute the rank of the matrix modulo each of these primes. Since the product of the primes necessarily exceeds the rth determinantal divisor, at least one of the primes does not divide it, and hence does not divide the largest non-zero invariant factor. The maximum rank obtained, then, is the correct rank. In practice, the

provisional rank has always been correct.

In the process of computing the rank, an $r \times r$ nonsingular submatrix is found. The determinant of this matrix can be computed modulo each prime in the previous list. The Chinese remainder theorem then establishes that the determinant is known exactly. In practice it is possible to easily obtain the determinant in a modular representation (see Cabay (1971) or Section 4.3.2 of Knuth (1969)). This value is a multiple of the rth determinantal divisor. There are two difficulties. Firstly, in cases where integer overflow interferes with the basic algorithm, the determinant is usually a very large number. This can be overcome by using a special package for handling large integers. The program uses the package of Brent (1978). Secondly, the determinant of just one $r \times r$ submatrix often provides a multiple which is orders of magnitude larger than the determinantal divisor.

While one determinant may provide too large a multiple of the rth determinantal divisor, the greatest common divisor of a small number of determinants of distinct $r \times r$ submatrices generally provides a reasonable multiple. The next section gives examples. The program's method of selection of a number of nonsingular $r \times r$ submatrices is such that the calculation of a number of determinants takes very little extra time over that taken to compute one determinant.

In general, the matrix determinant is much smaller than the Hadamard bound. Thus the determinant is calculated modulo more primes than is necessary. Cabay (1971) discusses this problem in relation to the exact solution of integer equations. He describes an early stopping criterion, called the *recursive test*, for reducing the number of primes modulo which the determinant must be calculated. There is an option in the program to use this criterion in the determinant calculation routine. There is no guarantee that the number obtained is not a proper divisor of the

determinant. However, except in specially constructed examples, this has never been the case.

Having found a multiple M of the rth determinantal divisor, its prime factorization, $M = \prod_{p|M} p^{\gamma(p)}$, must be found. It would be wrong to pass blithely over this step, because prime factorization of large numbers may be difficult, for example see Section 4.5.4 of Knuth (1969). However, the multiple produced by the program in its applications has always been good enough for factorization to pose no substantial difficulties. Further details of the implementation are described in Havas and Sterling (1979).

IV Applications and Program Performance

The initial stimulus which sparked the development of the program was a desire to investigate the Fibonacci group $F(2, 9)$ by a study of abelian quotients of its subgroups. Results of this inquiry are reported by Havas, Richardson and Sterling (1979), where background material may be found. We start off this section by looking at the abelian decomposition phase of the investigation of $F(2, 9)$.

$F(2, 9)$ may be presented with two generators and two relations, and has subgroups of index 2, 4, 8, 19, 38, 76 and 152, which are readily found. We denote by H_i a relation matrix for the maximal abelian quotient of one of these subgroups of index i, obtained in the following way.

Presentations for the subgroups themselves were found by finding subgroups of these indices in the maximal nilpotent quotient of $F(2, 9)$. Then the corresponding subgroups of $F(2, 9)$ itself, with the same index, were presented by the Reidemeister-Schreier program, denoted RS. The naive abelianization methods of RS were adequate to identify the maximal abelian quotients of subgroups of $F(2, 9)$ with index 2, 4 and 8, but

did not provide recognizable presentations for the maximal abelian quotients of subgroups of higher index.

In this section we first tabulate the performance of our algorithms on relation matrices associated with subgroups of $F(2, 9)$ with index exceeding 8 . For each index i subgroup of $F(2, 9)$, the presentation produced directly by the Reidemeister-Schreier method has $i + 1$ generators and $2i$ relations, so that the associated relation matrices have $2i$ rows and $i + 1$ columns. Because of initial problems handling H_{152} , caused by the size of the matrix, we went to the trouble of finding a better presentation for the (unique normal) index 152 subgroup by working down a chain of subgroups, in order to obtain a smaller matrix.

RS produced a 3 generator, 4 relation presentation for a subgroup of index 2 in $F(2, 9)$. Then a subgroup of index 4 in this subgroup was presented by RS on 9 generators and 16 relations, and this presentation was replaced by one on 4 generators and 9 relations, produced by a Tietze transformation program. From this presentation RS produced a 58 generator, 171 relation presentation for a subgroup of index 19 . We denote by $H_{2,4,19}$ a relation matrix for the maximal abelian quotient of this index 152 subgroup of $F(2, 9)$ obtained from this presentation. Finally we denote by H_{190} a relation matrix for the maximal abelian quotient of a subgroup of $F(2, 9)$ with index 190 , which was found as a consequence of calculations described in Havas, Richardson and Sterling (1979).

All results in this section are based on computer runs. We used a DEC KA10, with memory cycle time of 950 nanoseconds. Times quoted are in CPU seconds. Despite some variability due to the nature of DEC-10 timing methods, they provide a reasonable guide to relative performance. This machine was ideal for our purposes because it has a hardware integer

overflow check which is utilized by the FORTRAN operating system. Integers in FORTRAN on the DEC-10 are restricted to the range $-(2^{35}-1)$ to $2^{35}-1$.

Performance on Matrices Derived from $F(2, 9)$

	H_{19}	H_{38}	H_{76}	H_{152}	$H_{2,4,19}$	H_{190}
Rows	38	76	152	304	171	380
Columns	20	39	77	153	58	191
Torsion invariants	2,2	4	2	eighteen 5's	eighteen 5's	4
Eliminations (basic)	13	27	51	104	21	149
Time (basic)	1.3	4.9	22.1	108	20.1	133
Eliminations (modified)	20	39	75	124	48	146
Time (modified)	1.4	12.5	251	1457	492	2674
First determinant	19752	613568	6789296	$2^2.7^2.5^{20}$	$2^3.3^2.5^{19}$	$\sim 1 \times 10^{17}$
G.c.d.	8	32	$65536 = 2^{15}$	$2^2.7^2.5^{20}$	3.5^{19}	256
Primes (rec./Had.)	2/2	3/4	3/7	5/13	4/8	5/15
Time (Had., 1 det.)	2.4	13.3	80	724	81	1519
Time (rec., 1 det.)	2.4	11.1	45	346	46	627
Time (Had., g.c.d.)	5.2	17.5	88	937	90	1568
Time (rec., g.c.d.)	5.2	12.6	51	538	53	643
Time (modular)	1.4	6.2	34.5	192	45	336

Notes. (a) The top section of the table describes the nature of the matrices involved. In only one case, $H_{2,4,19}$, does an initial matrix include entries which exceed 1 in magnitude. $H_{2,4,19}$ has entries with

magnitude up to 3 . The torsion invariants reveal that no diagonal form has entries exceeding 5 .

(b) The second section describes the performance of the basic algorithm. Eliminations (basic) indicates the number of eliminations successfully performed (that is, the number of diagonal entries found) using the basic algorithm, prior to integer overflow occurring. Observe that the basic algorithm did not terminate successfully in any of these cases. Time (basic) gives the time till integer overflow.

(c) The third section describes the performance of the basic algorithm combined with the heuristic modifications, using default settings for the parameters. Eliminations (modified) indicates the number of eliminations successfully performed before integer overflow. Observe that both H_{19} and H_{38} were handled properly. Further, in the case of H_{76} , after 75 eliminations 10 overflows occurred and then, in just 3 more seconds, a correct diagonal form was attained, in spite of the overflows. Time (modified) gives the time till successful diagonalisation or overflow.

(d) The fourth section describes the performance of the determinant and greatest common divisor calculation routines. G.c.d. gives the greatest common divisor of the first 4 determinants, where available. In fact in all cases bar $H_{2,4,19}$ this greatest common divisor is attained from two determinants. For $H_{2,4,19}$ 3 determinants were used, with the first 2 providing a greatest common divisor a factor of 3 higher. Primes (rec./Had.) gives the number of primes required for determinant calculation, first using the recursive test, second using the Hadamard bound. Time (Had., 1 det.) indicates the time taken to compute 1 determinant using the Hadamard bound, and Time (Had., g.c.d.) indicates the time taken to compute 4 determinants and their greatest common divisor, using the Hadamard bound. The other times in this section are for calculations using

the recursive test. For H_{38} 3 primes were used to compute the first determinant using the recursive test, but when 4 determinants were calculated the fourth required only 2 primes, so the greatest common divisor time involved only two primes. For H_{152} only two non-singular submatrices involving the first 152 linearly independent rows were found, so only 2 determinants were involved in the greatest common divisor calculations. Finding maximal rank submatrices generally takes up a substantial part of the determinant calculation time.

(e) The final section provides the performance of the computation of the Smith normal form of matrices considered as $\mathbb{Z}_{(p)}$-matrices. In each case, this is for complete normal form calculation. The time taken does not depend significantly on either the prime or the power of the prime modulo which the calculation was done.

Generally speaking, this table indicates that the heuristic modifications extended the range of the basic algorithm. However for the larger matrices, the alternative approach using modular calculations was far superior.

Further examples of the program's performance are given by computations with four 26×27 matrices which arose from investigations of the fundamental groups of two 11-crossing knots. Details were given by Havas and Kovács (1979). In these cases no initial matrix entry exceeded 7 in magnitude, the rank was always 25 , and the torsion invariants were $\{3\}$, $\{14\}$, $\{2\}$, and $\{3, 3\}$ respectively.

Modular techniques readily identified the associated abelian groups; the greatest common divisors of 3 determinants were 15, 56, 6 and 18 respectively. However the basic algorithm was hopelessly inadequate and even with heuristic modifications some difficulties were encountered. Further specific details appear in Havas and Sterling (1979). Special

mention is made of the behaviour of the heuristic modifications.

Applications of the program have been made in identifying groups defined by fourth powers (see M.F. Newman (1976b)), in investigating link complements of knots in hyperbolic space (Grunewald (1980)), and in identifying the kernel of a homomorphism from

$$\langle X, Y; X^7 = Y^7 = (XY)^7 = \left(XYX^{-1}Y\right)^2 = \emptyset \rangle$$

to PSL(2, 7) . The size of matrices ranged from small in the first instance, through moderate (48 × 84 for a typical example in the second instance), to an initial matrix of 156 × 169 for the last application. In all cases however, the heuristically modified algorithm had no difficulty in computing the Smith normal form. On the other hand, following D.A. Smith (1966), we looked at some random matrices with single digit entries. A 13 × 13 matrix of this kind had determinant $2^4.3.13.7993.20175973$ and invariant factors {2, 50315164282968} . This calculation was done *via* the congruential methods.

CHAPTER 3

TORSIONFREE GROUPS

A particular presentation, called a canonical presentation, can be associated with each finitely generated torsionfree nilpotent group of class 2 . Canonical presentations are potentially a powerful tool for investigating torsionfree groups. They are used, in this chapter and the next, to classify all finitely generated torsionfree nilpotent groups of class 2 with Hirsch number less than or equal to 6 .

In the first section these special presentations are introduced and the main theorem is proved. This theorem gives conditions, in terms of integer matrices, for when two canonical presentations present isomorphic groups. The result is applied, in section II, to give some classification results, extending the work of Grunewald and Scharlau (1979). The final section introduces a polynomial invariant of torsionfree groups.

Throughout the chapter, all groups are finitely generated and nilpotent of class 2 .

I Canonical Presentations

LEMMA 3.1. *Let G be torsionfree. Then*

(i) $G/I(G')$ is torsionfree,

(ii) $I(G')$ is central.

Proof. *(i)* Suppose $a^m \in I(G')$. Then $(a^m)^n \in G'$, for some n , and $a \in I(G')$.

(ii) Let $a \in I(G')$, $b \in G$. Then $a^m \in G' \leq Z(G)$. So $[b, a^m] = \emptyset = [b, a]^m$. Since G is torsionfree, this implies $[b, a]$ is trivial. So $I(G')$ is central, and thus abelian. □

As a consequence of the above lemma, two invariants can be associated with each torsionfree group G, namely the ranks as free abelian groups of $G/I(G')$ and $I(G')$. These will be denoted d, s respectively. Given any presentation there is an algorithm to compute d and s. The details will be described in Chapter 6. Note that $1 \leq s \leq \binom{d}{2}$. The case $s = 0$ corresponds to a free abelian group of rank d, which does not have class 2. Also $h(G)$, the Hirsch number of G, equals $d + s$.

$\mathcal{T}(d, s)$ will denote the family of torsionfree groups with $G/I(G')$ a free abelian group of rank d, and $I(G')$ a free abelian group of rank s.

For the rest of this section G always belongs to $\mathcal{T}(d, s)$.

DEFINITION 3.2. Choose elements $a_1, \ldots, a_d$ of G such that $a_1 I(G'), \ldots, a_d I(G')$ form a basis of $G/I(G')$, and elements $b_1, \ldots, b_s$ that form a basis of $I(G')$. Then G can be presented by

$$\Big\langle a_1, \ldots, a_d, b_1, \ldots, b_s;\ [a_j, a_i] = \prod_{k=1}^{s} b_k^{\alpha(i,j,k)},\ 1 \leq i < j \leq d,$$

$$[b_j, b_i] = \emptyset,\ 1 \leq i \leq d,\ 1 \leq j \leq s,$$

$$[b_j, b_i] = \emptyset,\ 1 \leq i, j \leq s,\ \alpha(i, j, k) \in \mathbb{Z}\Big\rangle .$$

Such a presentation is called a *canonical* presentation, following the language of Grunewald and Segal (1979b).

In subsequent presentations, the term 'canonical' will imply the conditions $\alpha(i, j, k) \in \mathbb{Z}$, and the relations

$$[b_j, a_i] = \emptyset,\ 1 \leq i \leq d,\ 1 \leq j \leq s,\ [b_j, b_i] = \emptyset,\ 1 \leq i < j \leq s .$$

Note that $\{a_1, \ldots, a_d, b_1, \ldots, b_s\}$ is a canonical basis as defined by P. Hall (1969).

A given canonical presentation is determined by the $\alpha(i, j, k)$'s.

These parameters may be thought of as representing an alternating bilinear map α from $G/I(G') \times G/I(G')$ to $G' \leq I(G')$ given by $\alpha\{uI(G'), vI(G')\} = [u, v]$. Because of the bilinearity of the map, note Lemma 1.3, it is sufficient to specify the image of the generators. A skew-symmetric matrix whose (i, j)th entry is $[a_j, a_i]$ then represents α . Viewed in this way, the problem of classifying torsionfree class 2 nilpotent groups is a generalisation of the classical problem of classifying alternating bilinear forms on a vector space. This aspect of the thesis has been mentioned in the introduction, and will not be considered further here.

Consider the group with the canonical presentation of Definition 3.2. Every element of the group can be written uniquely in the form $\prod_{i=1}^{d} a_i^{\rho(i)} \prod_{k=1}^{s} b_k^{\sigma(k)}$, where $\rho(i)$, $\sigma(k)$ are integers for $1 \leq i \leq d$ and $1 \leq k \leq s$. A word in this form is called *normal*. An algorithm, called *collection*, is described which, when given a word in $\{a_1, \ldots, a_d, b_1, \ldots, b_s\}$ computes the normal word representing the same group element.

Consider a word in $\{a_1, \ldots, a_d, b_1, \ldots, b_s\}$.

1. Move all occurrences of b_s and b_s^{-1} to the right hand end of the word. Sum the exponents of b_s to give a word of the form $vb_s^{\sigma(s)}$, where b_s does not appear in v . This step is possible since b_s is central.

2. Proceed similarly for $b_{s-1}, b_{s-2}, \ldots, b_1$ to give a word of the form $w \prod_{k=1}^{s} b_k^{\sigma(k)}$, where w is a word in $\{a_1, \ldots, a_d\}$.

3. If w is a normal word, then the collection is finished. If w is not normal, then it has a subword in the following list:

   (i) $a_i^{-1}a_i$, $a_ia_i^{-1}$, $1 \le i \le d$,

   (ii) a_ja_i, $a_j^{-1}a_i^{-1}$, $1 \le i < j \le d$,

   (iii) $a_j^{-1}a_i$, $a_ja_i^{-1}$, $1 \le i < j \le d$.

4. If the subword is of type (i), delete the subword to give a word w' . Go to step 3 with $w' \prod_{k=1}^{s} b_k^{\sigma(k)}$. If the subword is of type (ii), replace a_ja_i and $a_j^{-1}a_i^{-1}$ by a_ia_j, $a_i^{-1}a_j^{-1}$ respectively to give a word w' . Go to step 3 with $w' \prod_{k=1}^{s} b_k^{\sigma'(k)}$, where $\sigma'(k) = \sigma(k) + \alpha(i, j, k)$. If the subword is of type (iii), replace $a_j^{-1}a_i$ and $a_ja_i^{-1}$ by $a_ia_j^{-1}$, $a_i^{-1}a_j$ respectively to give a word w' . Go to step 3 with $w' \prod_{k=1}^{s} b_k^{\sigma'(k)}$, where $\sigma'(k) = \sigma(k) - \alpha(i, j, k)$.

Steps 3 and 4 are repeated at most $n(n-1)/2$ times if n is the number of symbols in the original word.

This algorithm is suitable for both hand and machine calculation. It will be used implicitly in later calculations. Collection as described here is a particular example of a more general process described in more detail in M.F. Newman (1976a) and Havas and Nicholson (1976).

A skew-symmetric matrix can be associated with every canonical presentation. The entries of the matrix, which has dimensions $d \times d$, are linear homogeneous polynomials with integer coefficients in the s

indeterminates $x_1, \ldots, x_s$. It is convenient to let $\alpha(i, i, k) = 0$ and $\alpha(i, j, k) = -\alpha(j, i, k)$ for $i > j$.

Given a canonical presentation P as in Definition 3.2, the associated $d \times d$ skew symmetric matrix, denoted M_P, is defined by

$$M_P(i, j) = \sum_{k=1}^{s} \alpha(i, j, k)x_k .$$

The parameters $\alpha(i, j, k)$ clearly depend on the choice of the a_i's and b_j's in Definition 3.2. Thus there may be many canonical presentations for the same torsionfree group. However, it is possible to give conditions in terms of the associated skew-symmetric matrices for two canonical presentations to present isomorphic groups. The rest of this section states and proves this result.

Let $g_1, \ldots, g_d$ be elements of G such that $\{g_1 I(G'), \ldots, g_d I(G')\}$ is another basis of $G/I(G')$. By Theorem 1.2, there is an element T of $GL(d, \mathbb{Z})$ such that $g_i I(G') = \prod_{m=1}^{d} a_m I(G')^{T(i,m)}$. A new canonical presentation, $\overline{P}$, can be given for G in terms of the g_i's and b_j's. Further, $\overline{P}$ can be obtained from P by applying Tietze transformations in the manner of the example after Theorem 1.4. Another set of parameters $\beta(i, j, k)$ would be obtained. Then $M_{\overline{P}}(i, j) = \sum_{k=1}^{s} \beta(i, j, k)x_k$, where $\beta(i, i, k) = 0$ and $\beta(i, j, k) = -\beta(j, i, k)$ for $i > j$.

$$[g_j, g_i] = \left[\prod_{n=1}^{d} a_n^{T(j,n)}x, \prod_{m=1}^{d} a_m^{T(i,m)}y\right]$$

for some elements x, y in $I(G')$,

$$= \left[\prod_{n=1}^{d} a_n^{T(j,n)}, \prod_{m=1}^{d} a_m^{T(i,m)}\right] \text{ by Lemma 3.1,}$$

$$= \prod_{m=1}^{d}\prod_{n=1}^{d} [a_n, a_m]^{T(i,m)T(j,n)}$$

$$= \prod_{m=1}^{d}\prod_{n=1}^{d}\prod_{k=1}^{s} b_k^{\alpha(m,n,k)T(i,m)T(j,n)} .$$

Reordering the product gives that

$$\beta(i, j, k) = \sum_{m=1}^{d}\sum_{n=1}^{d} \alpha(m, n, k)T(i, m)T(j, n) ,$$

and so

$$M_P(i, j) = \sum_{k=1}^{s}\sum_{m=1}^{d}\sum_{n=1}^{d} \alpha(m, n, k)T(i, m)T(j, n)x_k .$$

Consider the matrix TM_PT^t. The (i, j)th entry of this matrix equals

$$\sum_{m=1}^{d} T(i, m)\sum_{n=1}^{d} M_P(m, n)T^t(n, j) = \sum_{m=1}^{d}\sum_{n=1}^{d} T(i, m)M_P(m, n)T(j, n)$$

$$= \sum_{m=1}^{d}\sum_{n=1}^{d}\sum_{k=1}^{s} \alpha(m, n, k)T(i, m)T(j, n)x_k .$$

So TM_PT^t equals M_P, the matrix associated with the canonical presentation relative to the bases $g_1I(G'), \ldots, g_dI(G')$ and $b_1, \ldots, b_s$.

Note that if M is a skew-symmetric matrix, so too is TMT^t.

The multiplication of matrices can be viewed as an action of $GL(d, \mathbb{Z})$ on the skew-symmetric matrices. If T is an elementary matrix, then the

action is to perform both the corresponding elementary row operation and the corresponding elementary column operation. The order in which these are done does not matter, as can be seen from the associativity of matrix multiplication.

Returning attention to the presentation of Definition 3.2, we consider the effect of a change of basis of $I(G')$. Let $h_1, \ldots, h_s$ be the elements of another basis of $I(G')$. By Theorem 1.2 there is an element S of $\mathrm{GL}(s, \mathbb{Z})$ such that

$$h_k = \prod_{L=1}^{s} b_L^{S(k,L)} .$$

Then

$$b_k = \prod_{L=1}^{s} h_L^{S^{-1}(k,L)} .$$

$$[a_j, a_i] = \prod_{k=1}^{s} b_k^{\alpha(i,j,k)}$$

$$= \prod_{k=1}^{s} \prod_{L=1}^{s} h_L^{S^{-1}(k,L)\alpha(i,j,k)} .$$

If $\tilde{P}$ is the canonical presentation of G relative to the elements $a_1, \ldots, a_d$ and $h_1, \ldots, h_s$, then $M_{\tilde{P}}(i, j) = \sum_{k=1}^{s} \gamma(i, j, k)x_k$, where

$$\gamma(i, j, k) = \sum_{L=1}^{s} S^{-1}(L, k)\alpha(i, j, L)$$

$$= \sum_{k=1}^{s} \sum_{L=1}^{s} \alpha(i, j, L)S^{-1}(L, k)x_k .$$

Note $\tilde{P}$ can be obtained from P by application of Tietze transformations.

There is an action of $\mathrm{GL}(s, \mathbb{Z})$ on $\mathbb{Z}[x_1, \ldots, x_s]$ induced by the linear substitutions $x_k \to \sum_{L=1}^{s} S(k, L)x_L$, for S an element of $\mathrm{GL}(s, \mathbb{Z})$.

Let M_P^S denote the matrix corresponding to this action of S in $GL(s, \mathbb{Z})$ on the entries of M_P. Thus

$$M_P^S(i, j) = \sum_{k=1}^{s} \sum_{L=1}^{s} \alpha(i, j, k) S(k, L) x_L .$$

So $M_P^{S^{-1}}$ equals $M_{\tilde{P}}$.

Combining the changes of bases gives the following. Let Q be the canonical presentation of G relative to the elements $g_1, \ldots, g_d$ and $h_1, \ldots, h_s$. If $M_Q = \sum_{k=1}^{s} \beta(i, j, k) x_k$, then

$$\beta(i, j, k) = \sum_{L=1}^{s} \sum_{m=1}^{d} \sum_{n=1}^{d} S^{-1}(L, k) T(i, m) T(j, n) \alpha(i, j, L) .$$

Note that $(TMT^t)^S = TM^S T^t$ because the order of the sum can be changed. The main theorem of this chapter can now be stated.

THEOREM 3.3. *Let P, Q be two canonical presentations. Then P, Q present isomorphic groups iff there exist elements S of $GL(s, \mathbb{Z})$ and T of $GL(d, \mathbb{Z})$ such that $M_Q = TM_P^S T^t$.*

Proof. Let

$$P = \left\langle a_1, \ldots, a_{d_1}, b_1, \ldots, b_{s_1}; [a_j, a_i] = \prod_{k=1}^{s} b_k^{\alpha(i,j,k)}, \right.$$
$$\left. 1 \le i < j \le d, \text{ canonical} \right\rangle$$

and

$$Q = \left\langle g_1, \ldots, g_{d_2}, h_1, \ldots, h_{s_2}; [h_j, h_i] = \prod_{k=1}^{s} h_k^{\beta(i,j,k)}, \right.$$
$$\left. 1 \le i < j \le d, \text{ canonical} \right\rangle$$

present G, H respectively.

IF By assumption $d_1 = d_2 = d$ and $s_1 = s_2 = s$. Also by the

calculations above

$$\beta(i, j, k) = \sum_{L=1}^{s} \sum_{m=1}^{d} \sum_{n=1}^{d} \alpha(i, j, L)S(L, k)T(i, m)T(j, n) . \qquad (1)$$

Let $g = \prod_{i=1}^{d} g_i^{\rho(i)} \prod_{k=1}^{s} h_k^{\sigma(k)}$ be an arbitrary element of H. Consider the map $\theta : H \to G$ given by

$$g\theta = \prod_{i=1}^{d} a_i^{\omega(i)\rho(i)} \prod_{k=1}^{s} b_k^{\eta(k)\sigma(k)} ,$$

where

$$\omega(i) = \sum_{m=1}^{d} T(i, m) \quad \text{and} \quad \eta(k) = \sum_{L=1}^{s} S^{-1}(L, k) .$$

Since S, T are invertible matrices, θ is 1-1 and onto.

Let

$$h = \prod_{i=1}^{d} g_i^{\mu(i)} \prod_{k=1}^{s} h_k^{\nu(k)} .$$

Then

$$gh = \prod_{i=1}^{d} g_i^{\rho(i)+\mu(i)} \prod_{k=1}^{s} b_k^{\sigma(k)+\nu(k)+\lambda(k)} ,$$

where

$$\lambda(k) = \sum_{1 \le i < j \le d} \rho(j)\mu(i)\beta(i, j, k) .$$

$$(gh)\theta = \prod_{i=1}^{d} a_i^{\omega(i)\left(\rho(i)+\mu(i)\right)} \prod_{k=1}^{s} b_k^{\eta(k)\left(\sigma(k)+\nu(k)+\lambda(k)\right)} ,$$

where $\omega(i)$, $\eta(k)$ and $\lambda(k)$ are as defined above.

Then $(gh)\theta = (g\theta)(h\theta)$ if

$$\eta(k)\lambda(k) = \sum_{1 \le i < j \le d} \sum_{L=1}^{s} S^{-1}(L, k)\rho(j)\mu(i)\beta(i, j, k)$$

equals $\sum_{1 \le i < j \le d} \sum_{m=1}^{d} \sum_{n=1}^{d} T(i, m)\mu(i)T(j, n)\rho(j)\alpha(i, j, k) .$

But equality holds by equation (1), and θ is an isomorphism.

ONLY IF Let $\theta : H \to G$ be an isomorphism. Then

$$d_1 = \text{rank of } G/I(G') = \text{rank of } H/I(H') = d_2 = d ,$$

and

$$s_1 = \text{rank of } I(G') = \text{rank of } I(H') = s_2 = s .$$

Since θ is an isomorphism $(g_1\theta)I(G'), \ldots, (g_d\theta)I(G')$ form a basis of $G/I(G')$ and $h_1\theta, \ldots, h_s\theta$ form a basis of $I(G')$. By Theorem 1.4, Tietze transformations can be applied to P to give a canonical presentation $\bar{P}$ for G in terms of the $g_i\theta$'s and $h_k\theta$'s. By the earlier discussion $M_{\bar{P}} = TM_P^S T^t$ for elements S of $\mathrm{GL}(s, \mathbb{Z})$ and T of $\mathrm{GL}(d, \mathbb{Z})$.

Because θ is an isomorphism,

$$[g_j\theta, g_i\theta] = [g_j, g_i]\theta = \left(\prod_{k=1}^{s} h_k^{\beta(i,j,k)}\right)\theta$$

$$= \prod_{k=1}^{s} (h_k\theta)^{\beta(i,j,k)} .$$

Thus the parameters of $\bar{P}$ are identical to the parameters of Q and the theorem is proved. □

II Some Classification Results

In this section, Theorem 3.3 is used to classify groups in $T(d, 1)$ and $T(3, s)$. Most of the results given were also essentially obtained by Grunewald and Scharlau (1979), who investigated torsionfree nilpotent groups of class 2 by considering their finite quotients. The corresponding propositions of their paper will be indicated.

Groups of $T(d, 1)$ are considered first. To classify them, a result about integer skew-symmetric matrices is needed.

PROPOSITION 3.4. *Let A be a $d \times d$ integer skew-symmetric matrix. Then there is an element T of $\mathrm{GL}(d, \mathbb{Z})$ such that*

$$TAT^t = \begin{bmatrix} 0 & h_1 & & & & \\ -h_1 & 0 & & & & \\ & & \ddots & & & \\ & & & 0 & h_r & \\ & & & -h_r & 0 & \\ & & & & & 0 \end{bmatrix},$$

where $h_i \geq 0$ and $h_{i-1} | h_i$, $2 \leq i \leq r = \lfloor d/2 \rfloor$.

This is a particular case of Theorem IV.1 of Morris Newman (1972). A proof is given in his book. Now $\begin{bmatrix} 0 & h_i \\ -h_i & 0 \end{bmatrix}$ is equivalent in the sense of integer matrices to $\begin{bmatrix} h_i & 0 \\ 0 & h_i \end{bmatrix}$. Thus the invariant factors of a skew-symmetric matrix occur in pairs. Computing the Smith normal form of a skew-symmetric matrix, then, will give the h_i's of the above proposition. In fact the basic algorithm described in Section 2.II can be modified to give a proof of the proposition. Furthermore a matrix T can be explicitly determined by recording the elementary operations performed. The following theorem is a generalisation of the essential part of Proposition C of Grunewald and Scharlau (1979).

THEOREM 3.5. *Let G belong to $T(d, 1)$. Then G has a presentation of the form*

$$\Big\langle a_1, \ldots, a_d, b; [a_2, a_1] = b^{h_1}, [a_4, a_3] = b^{h_2}, \ldots, [a_{2r}, a_{2r-1}] = b^{h_r},$$
$$[a_j, a_i] = \emptyset \text{ otherwise}, h_i \in \mathbb{Z}, h_1 > 0,$$
$$h_i \geq 0 \text{ and } h_{i-1} | h_i \text{ for } 2 \leq i \leq r = \lfloor d/2 \rfloor, \text{ canonical} \Big\rangle.$$

Presentations of this form with different values for $h_1, \ldots, h_r$ present

nonisomorphic groups.

Proof. Let P be a canonical presentation of a group in $\mathcal{T}(d, 1)$. Consider the $d \times d$ integer matrix M where $M(i, j)$ is the coefficient of x_1 in $M_P(i, j)$. By Proposition 3.4, there is an element T of $GL(d, \mathbb{Z})$ such that TMT^t has the special form. The canonical presentation with the associated skew-symmetric matrix TM_PT^t has the form as stated in the first part of the theorem. Let $r = \lfloor d/2 \rfloor$.

Let

$$P = \langle a_1, \ldots, a_d, b; [a_2, a_1] = b^{h_1}, [a_4, a_3] = b^{h_2}, \ldots, [a_{2r}, a_{2r-1}] = b^{h_r},$$

$$[a_j, a_i] = \emptyset \text{ otherwise}, h_1 > 0, h_i \geq 0 \text{ and } h_{i-1} | h_i, 2 \leq i \leq r, \text{ canonical} \rangle$$

and

$$Q = \langle a_1, \ldots, a_d, b; [a_2, a_1] = b^{k_1}, [a_4, a_3] = b^{k_2}, \ldots, [a_{2r}, a_{2r-1}] = b^{k_r},$$

$$[a_j, a_i] = \emptyset \text{ otherwise}, k_1 > 0, k_i \geq 0 \text{ and } k_{i-1} | k_i, 2 \leq i \leq r, \text{ canonical} \rangle$$

present G, H respectively.

Let j be the first index such that $h_j \neq k_j$. If no such j exists, then G and H are isomorphic. Assume without loss of generality that $k_j < h_j$. Then G/G^{h_j} is not isomorphic to H/H^{h_j}, by consideration of their abelian subgroups. □

The rest of this section gives a classification of all torsionfree nilpotent groups of class 2, with Hirsch number less than or equal to 6, apart from groups in $\mathcal{T}(4, 2)$ which are considered in the next chapter.

PROPOSITION 3.6. *Every torsionfree nilpotent group of class* 2 *and Hirsch number* 3 *has a presentation of the form*

$$\langle a_1, a_2, b; [a_2, a_1] = b^{\alpha}, [a_1, b] = \emptyset = [a_2, b], \alpha \in \mathbb{Z}^+ \rangle .$$

Different values of α *give nonisomorphic groups.*

Proof. Apply Theorem 3.5. □

This family of groups is used as an example in Pickel (1971).

PROPOSITION 3.7. *Every group in* $T(3, 1)$ *has a presentation of the form*

$$\langle a_1, a_2, a_3, b; [a_2, a_1] = b^{\alpha}, [a_3, a_1] = \emptyset = [a_3, a_2], \alpha \in \mathbb{Z}^+, [b, a_i] = \emptyset, 1 \leq i \leq 3 \rangle .$$

Different values of α *give nonisomorphic groups.*

Proof. Apply Theorem 3.5. □

Theorem 3.5 can be similarly used to write down explicit presentations for groups in $T(4, 1)$ and $T(5, 1)$. Both the next proposition, concerned with groups in $T(3, 2)$, and Proposition 4.1, related to $T(4, 2)$, give algorithms to be applied to skew-symmetric matrices. For a given matrix, M, the objective is to find matrices S, T such that TMS^tT^t has a particular, simple form. An equivalent aim, by Lemma 2.2, is to find a sequence of elementary matrices which transform the matrix into its particular form. The algorithms are described in this latter context.

The description of the action of the elementary matrices is consistent in the two propositions. Linear substitutions are given explicitly. The other action is defined by a specific row or column operation. For example, interchange columns i and j, or add q times row i to row j. Always the unspecified corresponding column or row operation is to be performed. The choice of operation specified is determined by which is more relevant to the particular element in the upper half of the matrix on which attention is focussed.

PROPOSITION 3.8. *Every group in* $T(3, 2)$ *has a presentation of the form*

$$\left\langle a_1, a_2, a_3, b_1, b_2; [a_2, a_1] = b_1^{\alpha}, [a_3, a_1] = b_2^{\beta}, \right.$$

$$\left. [a_3, a_2] = \emptyset, \alpha, \beta \in \mathbb{Z}^+, \alpha|\beta \text{ canonical} \right\rangle .$$

Different values of α, β *give nonisomorphic groups.*

Proof. Let G be in $T(3, 2)$. Choose a canonical presentation for G as in Definition 3.2. The associated skew-symmetric matrix has the form

$$\begin{bmatrix} 0 & \alpha(1,2,1)x_1 + \alpha(1,2,2)x_2 & \alpha(1,3,1)x_1 + \alpha(1,3,2)x_2 \\ -\alpha(1,2,1)x_1 - \alpha(1,2,2)x_2 & 0 & \alpha(2,3,1)x_1 + \alpha(2,3,2)x_2 \\ -\alpha(1,3,1)x_1 - \alpha(1,3,2)x_2 & -\alpha(2,3,1)x_1 - \alpha(2,3,2)x_2 & 0 \end{bmatrix} .$$

Throughout the algorithm, $\alpha(i, j, k)$ always refers to the current value, not the initial value, of the coefficient of x_k in the (i, j)th entry of the matrix. The flowchart related to the algorithm of Proposition 4.1 on page 62 also gives, with appropriate modifications to stages B, F, and H, an overview of the algorithm to be described here. In particular, the claims establishing that the procedure is an algorithm apply. The table below gives the correspondence between the flowchart and this algorithm.

Stage	Steps	Objective of Stage
A	1-3	Ensure $\alpha(1, 2, 1) \neq 0$
B	4-7	$\alpha(1, 3, 1) = 0 = \alpha(2, 3, 1)$
C	8	$x_1 \to x_1 - qx_2$ where $\alpha(1, 2, 2) = q\alpha(1, 2, 1) + r$, $\lvert r\rvert < \lvert\alpha(1, 2, 1)\rvert$
D	9	Does $\alpha(1, 2, 2) = 0$?
E	10	Ensure $\alpha(1, 3, 2) \neq 0$
F	11-12	$\alpha(2, 3, 2) = 0$
G	13	Does $\alpha(1, 2, 1) \mid \alpha(1, 3, 2)$?
H	14-15	$\alpha(1, 2, 1), \alpha(1, 3, 2) > 0$

The steps of the algorithm are:-

1. If $\alpha(1, 2, 1) \neq 0$, go to 4.

2. If $\alpha(1, 3, 1) = 0$, go to 3.

   Else, interchange columns 2 and 3, and go to 4.

3. Interchange rows 1 and 3.

4. Subtract q times column 2 from column 3, where

   $\alpha(1, 3, 1) = q\alpha(1, 2, 1) + r$ and $|r| < |\alpha(1, 2, 1)|$.

5. If $\alpha(1, 3, 1) = 0$, go to 6.

   Else interchange columns 2 and 3, and go to 4.

6. Add q times row 1 to row 3, where

   $\alpha(2, 3, 1) = q\alpha(1, 2, 1) + r$ and $|r| < |\alpha(1, 2, 1)|$.

7. If $\alpha(2, 3, 1) = 0$, go to 8.

   Else, interchange rows 1 and 3, and go to 4.

8. Send $x_1 \to x_1 - qx_2$, $x_2 \to x_2$, where

   $\alpha(1, 2, 2) = q\alpha(1, 2, 1) + r$ and $|r| < |\alpha(1, 2, 1)|$.

9. If $\alpha(1, 2, 2) = 0$, go to 10.

   Else send $x_1 \to x_2$, $x_2 \to x_1$, and go to 4.

10. If $\alpha(1, 3, 2) = 0$, interchange rows 1 and 2.

11. Subtract q times row 1 from row 2, where

    $\alpha(2, 3, 2) = q\alpha(1, 3, 2) + r$ and $|r| < |\alpha(1, 3, 2)|$.

12. If $\alpha(2, 3, 2) = 0$, go to 13.

    Else, interchange rows 1 and 2, and go to 11.

13. If $\alpha(1, 2, 1) | \alpha(1, 3, 2)$, go to 14.

    Else, send $x_1 \to x_1$, $x_2 \to x_1 + x_2$, and go to 4.

14. If $\alpha(1, 2, 1) < 0$, multiply column 2 by -1 .

15. If $\alpha(1, 3, 2) < 0$, multiply column 3 by -1 .

A canonical presentation associated with the current matrix has the form stated in the proposition, where $\alpha = \alpha(1, 2, 1)$ and $\beta = \alpha(1, 3, 2)$.

Clearly $I(G')/G' \cong C_\alpha \times C_\beta$ where $\alpha|\beta$. Since this abelian section is a group invariant different values of α, β give nonisomorphic groups. □

Proposition 3.8 is proved in Proposition E of Grunewald and Scharlau (1979). They use properties of the action of $GL(3, \mathbb{Z})$ on related 3×2 integer matrices to apply methods similar to those of Chapter 2. These methods would be much more efficient than the algorithm of Proposition 3.8 to calculate the isomorphism type of a group in $T(3, 2)$.

PROPOSITION 3.9. *Every group in* $T(3, 3)$ *has a presentation of the form*

$$\langle a_1, a_2, a_3, b_1, b_2, b_3; [a_2, a_1] = b_1^\alpha, [a_3, a_1] = b_2^\beta,$$
$$[a_3, a_2] = b_3^\gamma, \alpha, \beta, \gamma \in \mathbb{Z}^+, \alpha|\beta|\gamma, \textit{canonical}\rangle .$$

Different values of α, β, γ *give nonisomorphic groups.*

Proof. It is straightforward to modify the algorithm of Proposition 3.8 to prove the first statement. Let G be a group with the above presentation. Then

$$I(G')/G' \cong C_\alpha \times C_\beta \times C_\gamma \text{ where } \alpha|\beta|\gamma .$$

Since this abelian section is a group invariant, the proposition follows. □

Propositions 3.7, 3.8 and 3.9 constitute a classification of groups in $T(3, s)$, $1 \le s \le 3$.

III The Pfaffian

This section establishes an invariant of torsionfree groups, derived here from the skew-symmetric matrices associated with the canonical presentations described in the first section. This invariant, known as the Pfaffian, is only useful when the rank, d, of $G/I(G')$ is even. In this case it is an equivalence class of homogeneous polynomials of degree $d/2$ in s indeterminates.

The Pfaffian is introduced here naively. A more algebraic derivation of the Pfaffian can be distilled from Chapitre IX of Bourbaki (1959). Scheuneman (1967) introduces an analogous polynomial invariant for 2-step nilpotent Lie algebras via another approach.

Throughout this section A denotes a $d \times d$ skew-symmetric matrix. Then A is said to be of *odd (even) order* depending whether d is odd (even). The invariant to be considered is connected with the determinant of A.

PROPOSITION 3.10. *The determinant of a skew-symmetric matrix of odd order is zero.*

Proof. The property of skew-symmetry gives that A^t is obtained from A by multiplying each row (or column) by -1, whence

$$\det A = \det A^t = (-1)^d \det A .$$

Since d is odd, $\det A = 0$. □

The determinant of a skew-symmetric matrix of even order is well-known to be a square of a function of its elements. Some examples are considered.

When $d = 2$, $A = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix}$, and $\det A = a^2$. When $d = 4$,

$$A = \begin{bmatrix} 0 & a & b & c \\ -a & 0 & d & e \\ -b & -d & 0 & f \\ -c & -e & -f & 0 \end{bmatrix} .$$

$$\det A = a(af^2 - bef + cdf) - b(aef - be^2 + cde) + c(adf - bde + cd^2)$$

$$= (af - be + cd)^2 .$$

Let A^{ij}_{kl} denote the matrix obtained by deleting the ith and jth rows and kth and lth columns of A.

The *Pfaffian* of A, denoted $\mathrm{Pf}(A)$, is defined recursively as follows.

$$\mathrm{Pf}([0]) = 0 \ , \quad \mathrm{Pf}\left(\begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix}\right) = a \ ,$$

$$\mathrm{Pf}(A) = \sum_{j=2}^{d} (-1)^j A(1, j)\mathrm{Pf}\left(A_{1j}^{1j}\right) .$$

Note that if A is of odd order, then $\mathrm{Pf}(A) = 0$.

If A is the 4×4 matrix given earlier, then

$$\mathrm{Pf}(A) = (-1)^2.a.\mathrm{Pf}\left(\begin{bmatrix} 0 & -f \\ -f & 0 \end{bmatrix}\right) + (-1)^3.b.\mathrm{Pf}\left(\begin{bmatrix} 0 & e \\ -e & 0 \end{bmatrix}\right) + (-1)^4.c.\mathrm{Pf}\left(\begin{bmatrix} 0 & d \\ -d & 0 \end{bmatrix}\right)$$

$$= af - be + cd \ .$$

PROPOSITION 3.11. $\mathrm{Pf}(A)^2 = \det(A)$.

Proof. The essential structure of the proof dates back to Cauchy and is reported in Muir (1911).

The proposition is trivially true if A is of odd order. Proceed by induction on d when A is of even order. From above, the proposition is true for $d = 2$ and $d = 4$.

$$\mathrm{Pf}(A)^2 = \left[\sum_{j=2}^{d} (-1)^j A(1, j)\mathrm{Pf}\left(A_{1j}^{1j}\right)\right]^2$$

$$= \sum_{j=2}^{d} \left[A(1, j)^2 \mathrm{Pf}\left(A_{1j}^{1j}\right)^2 + 2 \sum_{k=j}^{d} (-1)^{j+k} A(1, j)A(1, k)\mathrm{Pf}\left(A_{1j}^{1j}\right)\mathrm{Pf}\left(A_{1k}^{1k}\right)\right] .$$

Expanding the determinant by the first row and first column gives

$$\det(A) = \sum_{j=2}^{d} \sum_{k=2}^{d} (-1)^{j+k-1} A(1, j)A(k, 1)\det\left(A_{1k}^{1j}\right) .$$

By skew-symmetry $A(1, j) = -A(j, 1)$.

Also $A_{1k}^{1j} = -\left(A_{1j}^{1k}\right)^t$ and hence $\det\left(A_{1k}^{1j}\right) = \det\left(A_{1j}^{1k}\right)$. Thus

$$\det(A) = \sum_{j=2}^{d} \left[A(1, j)^2 \det\left(A_{1j}^{1j}\right) + 2 \sum_{k=j}^{d} (-1)^{j+k} A(1, j)A(1, k)\det\left(A_{1j}^{1k}\right)\right] .$$

By induction

$$\det\left(A_{1j}^{1j}\right) = \mathrm{Pf}\left(A_{1j}^{1j}\right)^2 .$$

Thus it must be shown that $\det\left(A_{1j}^{1k}\right) = \mathrm{Pf}\left(A_{1j}^{1j}\right)\mathrm{Pf}\left(A_{1k}^{1k}\right)$. But this follows similarly by induction by expanding $\det\left(A_{1j}^{1k}\right)$ along its $(k\text{-}2)$th row and $(j\text{-}1)$th column. Note that the cofactor of $A_{1j}^{1k}(k\text{-}2, j\text{-}1)$ is a skew-symmetric matrix of odd order and hence zero. □

For the particular application of the next chapter, namely when $d = 4$, Proposition 3.11 has been proved by direct calculation.

Let M be a skew-symmetric matrix associated with a canonical presentation of a group in $T(d, s)$, where d is even. The entries of M are homogeneous linear polynomials in s indeterminates. It follows from the definition of the Pfaffian that $\mathrm{Pf}(M)$ is a homogeneous polynomial of degree $d/2$ in s indeterminates. Let S be an element of $\mathrm{GL}(s, \mathbb{Z})$. Then S acts on $\mathrm{Pf}(M)$ by linear substitution. The polynomial obtained by this action is denoted as $\mathrm{Pf}(M)S$.

THEOREM 3.12. *Let P, Q be two canonical presentations of groups in $T(d, s)$. If P, Q present isomorphic groups, then $\mathrm{Pf}(M_Q) = \pm\mathrm{Pf}(M_P)S$, for some element S in $\mathrm{GL}(s, \mathbb{Z})$.*

Proof. By Theorem 3.3, there exist elements S of $\mathrm{GL}(s, \mathbb{Z})$ and T of $\mathrm{GL}(d, \mathbb{Z})$ such that $M_Q = TM_P^S T^t$. Then

$$\begin{aligned}
\mathrm{Pf}(M_Q)^2 &= \det(M_Q) \quad \text{by Proposition 3.11,} \\
&= \det\left(TM_P^S T^t\right) \\
&= (\det T)^2 \det\left(M_P^S\right) \\
&= \mathrm{Pf}\left(M_P^S\right)^2 = \left(\mathrm{Pf}(M_P)S\right)^2 .
\end{aligned}$$

The theorem follows on taking square roots. □

CHAPTER 4

$T(4, 2)$

The classification of finitely generated nilpotent groups of class 2 and Hirsch number less than or equal to 6 is completed in this chapter with the discussion of groups in $T(4, 2)$. This family of groups is, in some sense, the first 'difficult' case for torsionfree nilpotent groups. For example, Grunewald and Scharlau (1979) give examples of nonisomorphic groups in $T(4, 2)$ which cannot be distinguished by their finite quotients. The classification results here are perhaps not as concise as the results of Section 3.II. However they enable many questions about these groups to be answered.

It is proved that groups in $T(4, 2)$ are identified by their maximal abelian quotient group, and by the equivalence class of a binary quadratic form, associated with a canonical presentation. The equivalence relation is a little different from the classical equivalence of binary quadratic forms under the action of $SL(2, \mathbb{Z})$.

The invariants are described with regard to a special type of canonical presentation which is introduced in the first section. The second section discusses the binary quadratic forms associated with the presentation. It also establishes that the two invariants determine a group in $T(4, 2)$ up to isomorphism. The next three sections discuss binary quadratic forms with respect to a special equivalence relation. An algorithm is given to determine when two binary quadratic forms are thus equivalent. Combining the methods of Section I leads to an algorithm for solving the isomorphism problem for two groups in $T(4, 2)$ given by canonical presentations. This algorithm is much more amenable for hand or machine calculation than the general algorithm of Grunewald and Segal (1979b) for solving the isomorphism

problem for finitely presented nilpotent groups. The final section gives some examples of how the methods can be applied to particular groups in $T(4, 2)$.

I. Restricted Canonical Presentations

The manipulation of skew-symmetric matrices which led, in the last chapter, to a normal form for groups in $T(d, 1)$, $T(3, 2)$ and $T(3, 3)$ can also be applied in this case of $T(4, 2)$. Here additional techniques are needed to characterise the groups. However, the matrix manipulation is a key step in the results that follow.

PROPOSITION 4.1. *The skew-symmetric matrix associated with a group in* $T(4, 2)$ *can be chosen to have the form*

$$\begin{bmatrix} 0 & \alpha x_1 & \beta x_2 & 0 \\ -\alpha x_1 & 0 & 0 & -\varepsilon x_2 \\ -\beta x_1 & 0 & 0 & \gamma x_1 + \delta x_2 \\ 0 & \varepsilon x_2 & -\gamma x_1 - \delta x_2 & 0 \end{bmatrix}$$

where $\alpha, \beta > 0$, $\gamma, \delta \geq 0$, $\alpha | \beta, \gamma$, $\beta | \delta, \varepsilon$.

Proof. The proof is similar to that of Proposition 3.8. An algorithm is given to change an arbitrary skew-symmetric matrix associated with a canonical presentation of a group in $T(4, 2)$ which has the form

$$\begin{bmatrix} 0 & \alpha(1,2,1)x_1 + \alpha(1,2,2)x_2 & \alpha(1,3,1)x_1 + \alpha(1,3,2)x_2 & \alpha(1,4,1)x_1 + \alpha(1,4,2)x_2 \\ -\alpha(1,2,1)x_1 - \alpha(1,2,2)x_2 & 0 & \alpha(2,3,1)x_1 + \alpha(2,3,2)x_2 & \alpha(2,4,1)x_1 + \alpha(2,4,2)x_2 \\ -\alpha(1,3,1)x_1 - \alpha(1,3,2)x_2 & -\alpha(2,3,1)x_1 - \alpha(2,3,2)x_2 & 0 & \alpha(3,4,1)x_1 + \alpha(3,4,2)x_2 \\ -\alpha(1,4,1)x_1 - \alpha(1,4,2)x_2 & -\alpha(2,4,1)x_1 - \alpha(2,4,2)x_2 & -\alpha(3,4,1)x_1 - \alpha(3,4,2)x_2 & 0 \end{bmatrix}$$

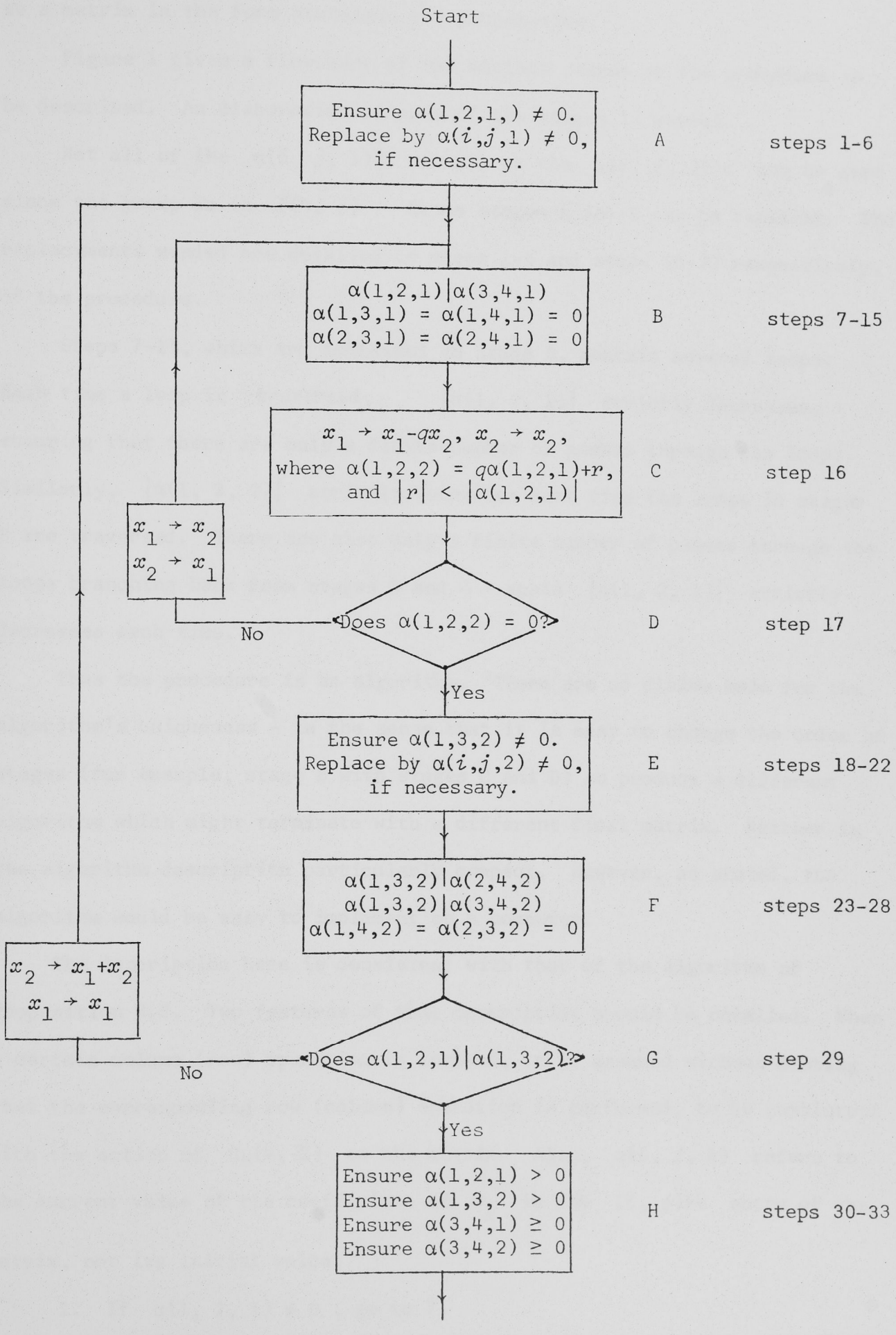

Start
Ensure α(1,2,1,) ≠ 0. Replace by α(i,j,1) ≠ 0, if necessary.
A
steps 1-6
α(1,2,1)|α(3,4,1)
α(1,3,1) = α(1,4,1) = 0
α(2,3,1) = α(2,4,1) = 0
B
steps 7-15
x1 → x1-qx2, x2 → x2,
where α(1,2,2) = qα(1,2,1)+r,
and |r| < |α(1,2,1)|
C
step 16
x1 → x2
x2 → x1
Does α(1,2,2) = 0?
No
D
step 17
Yes
Ensure α(1,3,2) ≠ 0. Replace by α(i,j,2) ≠ 0, if necessary.
E
steps 18-22
α(1,3,2)|α(2,4,2)
α(1,3,2)|α(3,4,2)
α(1,4,2) = α(2,3,2) = 0
F
steps 23-28
x2 → x1+x2
x1 → x1
Does α(1,2,1)|α(1,3,2)?
No
G
step 29
Yes
Ensure α(1,2,1) > 0
Ensure α(1,3,2) > 0
Ensure α(3,4,1) ≥ 0
Ensure α(3,4,2) ≥ 0
H
steps 30-33
Stop

FIGURE 1.

to a matrix in the form stated in the proposition.

Figure 1 gives a flowchart of the various stages of the procedure to be described. An elaboration of some of the stages is useful.

Not all of the $\alpha(i, j, 1)$'s nor all of the $\alpha(i, j, 2)$'s can be zero since the group is in $T(4, 2)$. Hence stages A and E can be realised. The replacements needed are detailed in steps 1-6 and steps 18-22 respectively, of the procedure.

Steps 7-15, which are contained in stage B, contain several loops. Each time a loop is traversed, $|\alpha(1, 2, 1)|$ strictly decreases, ensuring that there are only a finite number of passes through the loops. Similarly, $|\alpha(1, 3, 2)|$ strictly decreases each time the loops in stage F are traversed. There are also only a finite number of passes through the loops branching back from stages D and G - again $|\alpha(1, 2, 1)|$ strictly decreases each time.

Thus the procedure is an algorithm. There are no claims made for the algorithm's uniqueness - in the sense that it is easy to change the order of stages (for example, stage B with stages C and D) to produce a different algorithm which might terminate with a different final matrix. Neither is the algorithm description particularly compact. However, as stated, the algorithm would be easy to implement on a computer.

The description here is consistent with that of the algorithm of Proposition 3.8. Two features of that description should be recalled. When a certain column (row) operation is invoked, it is assumed without stating that the corresponding row (column) operation is performed, to be consistent with the action of $GL(4, \mathbb{Z})$ on the matrix. Also, $\alpha(i, j, k)$ refers to the current value of the coefficient of x_k in the (i, j)th entry of the matrix, not its initial value.

1. If $\alpha(1, 2, 1) \neq 0$, go to 7.

2. If $\alpha(1, 3, 1) = 0$, go to 3.

Else, interchange columns 2 and 3, and go to 7.

3. If $\alpha(1, 4, 1) = 0$, go to 4.

   Else, interchange columns 2 and 4, and go to 7.

4. If $\alpha(2, 3, 1) = 0$, go to 5.

   Else, interchange rows 1 and 3, and go to 7.

5. If $\alpha(2, 4, 1) = 0$, go to 6.

   Else, interchange rows 1 and 4, and go to 7.

6. Interchange rows 1 and 3, and columns 2 and 4, replacing $\alpha(1, 2, 1)$ by $\alpha(3, 4, 1)$. (If $\alpha(3, 4, 1)$ were zero the associated group would not be in $T(4, 2)$.)

7. Subtract q times column 2 from column 3, where $\alpha(1, 3, 1) = q\alpha(1, 2, 1) + r$, $|r| < |\alpha(1, 2, 1)|$.

8. If $\alpha(1, 3, 1) = 0$, go to 9.

   Else, interchange columns 2 and 3, and go to 7.

9. Subtract q times column 2 from column 4, where $\alpha(1, 4, 1) = q\alpha(1, 2, 1) + r$, $|r| < |\alpha(1, 2, 1)|$.

10. If $\alpha(1, 4, 1) = 0$, go to 11.

    Else, interchange columns 2 and 4, and go to 7.

11. Add q times row 1 to row 3, where $\alpha(2, 3, 1) = q\alpha(1, 2, 1) + r$, $|r| < |\alpha(1, 2, 1)|$.

12. If $\alpha(2, 3, 1) = 0$, go to 13.

    Else, interchange rows 1 and 3, and go to 7.

13. Add q times row 1 to row 4, where $\alpha(2, 4, 1) = q\alpha(1, 2, 1) + r$, $|r| < |\alpha(1, 2, 1)|$.

14. If $\alpha(2, 4, 1) = 0$, go to 15.

    Else, interchange rows 1 and 4, and go to 7.

15. If $\alpha(1, 2, 1) | \alpha(3, 4, 1)$, go to 16.

    Else, add row 3 to row 1, and go to 10.

The algorithm up to this stage has only considered the coefficients of

x_1 in the matrix. It is effectively an algorithm for computing the normal form for a matrix associated with a group in $T(4, 1)$.

16. Send $x_1 \to x_1 - qx_2$, $x_2 \to x_2$, where

$\alpha(1, 2, 2) = q\alpha(1, 2, 1) + r$, $|r| < |\alpha(1, 2, 1)|$.

17. If $\alpha(1, 2, 2) = 0$, go to 18.

Else, send $x_1 \to x_2$, $x_2 \to x_1$, and go to 7.

18. If $\alpha(1, 3, 2) \neq 0$, go to 23.

19. If $\alpha(1, 4, 2) = 0$, go to 20.

Else, interchange columns 3 and 4, and go to 23.

20. If $\alpha(2, 3, 2) = 0$, go to 21.

Else, interchange rows 1 and 2, and go to 23.

21. If $\alpha(2, 4, 2) = 0$, go to 22.

Else, interchange rows 1 and 2, and columns 3 and 4, and go to 23.

22. This step replaces $\alpha(1, 3, 2)$ by $\alpha(3, 4, 2)$, which is necessarily non-zero. It is more complicated than other replacement steps due to divisibility considerations. If $\alpha(1, 2, 1) | \alpha(3, 4, 2)$, add row 4 to row 1. Add $\alpha(3, 4, 1)/\alpha(1, 2, 1)$ times column 2 to column 3, and go to 30.

Else, send $x_2 \to x_1 + x_2$, $x_1 \to x_1$, and go to 15.

23. Subtract q times column 3 from column 4, where

$\alpha(1, 4, 2) = \alpha(1, 3, 2) + r$, $|r| < |\alpha(1, 3, 2)|$.

24. If $\alpha(1, 4, 2) = 0$, go to 25.

Else, interchange columns 3 and 4, and go to 23.

25. Subtract q times row 1 from row 2, where

$\alpha(2, 3, 2) = q\alpha(1, 3, 2) + r$, $|r| < |\alpha(1, 3, 2)|$.

26. If $\alpha(2, 3, 2) = 0$, go to 27.

Else, interchange rows 1 and 2, and go to 23.

27. If $\alpha(1, 3, 2) | \alpha(2, 4, 2)$, go to 28.

Else, add row 2 to row 1, and go to 23.

28. If $\alpha(1, 3, 2) | \alpha(3, 4, 2)$, go to 29.

Else, add row 3 to row 1. Subtract $\alpha(3, 4, 1)/\alpha(1, 2, 1)$ times column 2 from column 4, and go to 23.

29. If $\alpha(1, 2, 1) | \alpha(1, 3, 2)$, go to 30.

Else, send $x_2 \to x_1 + x_2$, $x_1 \to x_1$, and go to 7.

Before step 30, the matrix has the form

$$\begin{bmatrix} 0 & \alpha(1, 2, 1)x_1 & \alpha(1, 3, 2)x_2 & 0 \\ -\alpha(1, 2, 1)x_1 & 0 & 0 & \alpha(2, 4, 2)x_2 \\ -\alpha(1, 3, 2)x_2 & 0 & 0 & \alpha(3, 4, 1)x_1 + \alpha(3, 4, 2)x_2 \\ 0 & -\alpha(2, 4, 2)x_2 & -\alpha(3, 4, 1)x_1 - \alpha(3, 4, 2)x_2 & 0 \end{bmatrix},$$

where $\alpha(1, 2, 1), \alpha(1, 3, 2) \neq 0$. Also $\alpha(1, 2, 1) | \alpha(1, 3, 2), \alpha(3, 4, 1)$ and $\alpha(1, 3, 2) | \alpha(2, 4, 2), \alpha(3, 4, 2)$.

30. If $\alpha(1, 2, 1) < 0$, multiply column 2 by -1 .

31. If $\alpha(1, 3, 2) < 0$, multiply column 3 by -1 .

32. If $\alpha(3, 4, 1) < 0$, multiply column 4 by -1 .

33. If $\alpha(3, 4, 2) < 0$, multiply columns 3 and 4 by -1 .

Send $x_2 \to -x_2$ and $x_1 \to x_1$.

The matrix now has the required form, where $\alpha = \alpha(1, 2, 1)$, $\beta = \alpha(1, 3, 2)$, $\gamma = \alpha(3, 4, 1)$, $\delta = \alpha(3, 4, 2)$ and $\varepsilon = -\alpha(2, 4, 2)$. □

DEFINITION 4.2. A *restricted canonical presentation* is one of the form

$$\Big\langle a_1, a_2, a_3, a_4, b_1, b_2;\ [a_2, a_1] = b_1^{\alpha},\ [a_3, a_1] = b_2^{\beta},$$

$$[a_4, a_1] = \emptyset = [a_3, a_2],\ [a_4, a_2] = b_2^{-\varepsilon},\ [a_4, a_3] = b_1^{\gamma} b_2^{\delta},$$

$$[b_j, a_i] = \emptyset,\ 1 \le i \le 4,\ j = 1, 2,\ [b_2, b_1] = \emptyset,$$

$$\alpha, \beta, \gamma, \delta, \varepsilon \in \mathbb{Z},\ \alpha, \beta > 0,\ \alpha | \beta, \gamma,\ \beta | \delta, \varepsilon \Big\rangle .$$

Clearly a restricted canonical presentation presents a group in $T(4, 2)$. Conversely by Proposition 4.1, every group in $T(4, 2)$ has a restricted canonical presentation with the added properties that $\gamma, \delta \ge 0$.

The term 'restricted canonical' in a presentation will imply the relations $[b_j, a_i] = \emptyset$, $1 \le i \le 4$, $j = 1, 2$, $[b_2, b_1] = \emptyset$, and the conditions $\alpha, \beta, \gamma, \delta, \varepsilon \in \mathbb{Z}$, $\alpha, \beta > 0$ and $\alpha | \beta, \gamma$ and $\beta | \delta, \varepsilon$.

II. Invariants

In general, there are infinitely many different restricted canonical presentations for a given group. For example,

$$P_k = \Big\langle a_1, a_2, a_3, a_4, b_1, b_2;\ [a_2, a_1] = b_1,\ [a_3, a_1] = b_2,$$

$$[a_4, a_1] = [a_3, a_2] = \emptyset,\ [a_4, a_2] = b_2^{-k^2},$$

$$[a_4, a_3] = b_1 b_2^{2k},\ \text{restricted canonical} \Big\rangle$$

presents the same group for all integer values of k. This will be proved later in the chapter. The objective of this section is to give a set of invariants, immediately calculable from a restricted canonical presentation, which determines the isomorphism type of the group.

The isomorphism type of the abelian group $I(G')/G'$ is clearly an invariant of a group G in $T(4, 2)$. From the presentation of Definition 4.2, $I(G')/G' \cong C_\alpha \times C_\beta$ where $\alpha | \beta$. Thus α, β are invariants of G.

The ratio β/α is then also an invariant, which will be denoted λ. Note that this group is just the torsion subgroup of the maximal abelian quotient group.

Recall from section 3.III that the Pfaffian of a 4×4 skew-symmetric matrix M is $M(2, 1)M(4, 3) - M(3, 1)M(4, 2) + M(4, 1)M(3, 2)$. Thus for a matrix in the form of Proposition 4.1,

$$\begin{aligned} \mathrm{Pf}(M) &= \alpha x_1 . (\gamma x_1 + \delta x_2) - \beta x_2 . -\varepsilon x_2 + 0 \\ &= \alpha\gamma x_1^2 + \alpha\delta x_1 x_2 + \beta\varepsilon x_2^2 , \end{aligned}$$

which is a binary quadratic form.

Let $x = \alpha x_1$ and $y = \beta x_2$. Then

$$\mathrm{Pf}(M) = (\gamma/\alpha)x^2 + (\delta/\beta)xy + (\varepsilon/\beta)y^2 .$$

Note that the bracketed terms are all integers, because of the conditions on $\alpha, \beta, \gamma, \delta, \varepsilon$.

Thus a binary quadratic form is related to each restricted canonical presentation via the Pfaffian of the associated skew-symmetric matrix. Mildly abusing language, the *Pfaffian* of a restricted canonical presentation P, denoted $\mathrm{Pf}(P)$, is defined to be $\gamma' x^2 + \delta' xy + \varepsilon' y^2$, where P is in the form of Definition 4.2 and $\gamma' = \gamma/\alpha$, $\delta' = \delta/\beta$ and $\varepsilon' = \varepsilon/\beta$.

Some notation is introduced.

Let $f = \gamma x^2 + \delta xy + \varepsilon y^2$ be a binary quadratic form. The *discriminant* of f, denoted $\Delta(f)$, equals $\delta^2 - 4\gamma\varepsilon$. The *order* of f, denoted $o(f)$, is the greatest common divisor of γ, δ and ε. If $o(f) = 1$, then f is *primitive*. Also $(\gamma, \delta, \varepsilon)$ will be used as an abbreviation for $\gamma x^2 + \delta xy + \varepsilon y^2$.

Let $S = \begin{bmatrix} t & u \\ v & w \end{bmatrix}$ be an element of $\mathrm{GL}(2, \mathbb{Z})$. Then S acts naturally on binary quadratic forms by linear substitution. Thus S takes

$f = (\gamma, \delta, \varepsilon)$ to $fS = (\bar{\gamma}, \bar{\delta}, \bar{\varepsilon})$, where

$$fS = \gamma(tx+uy)^2 + \delta(tx+uy)(vx+wy) + \varepsilon(vx+wy)^2 .$$

The expressions for $\bar{\gamma}$, $\bar{\delta}$, and $\bar{\varepsilon}$ will be referred to frequently and so are labelled.

EQUATIONS 4.3. $$\bar{\gamma} = t^2\gamma + tv\delta + v^2\varepsilon ,$$

$$\bar{\delta} = 2tu\gamma + (tw+uv)\delta + 2vw\varepsilon ,$$

$$\bar{\varepsilon} = u^2\gamma + uw\delta + w^2\varepsilon .$$

Let $GL_\lambda(2, \mathbb{Z})$ denote the subgroup of $GL(2, \mathbb{Z})$ consisting of elements of the form $\begin{bmatrix} t & u \\ v\lambda & w \end{bmatrix}$. The corresponding subgroups of the modular group have arisen in classical contexts - see, for example, Fricke and Klein (1897). Morris Newman (1972) discusses generalisations of these groups in some detail in Chapter VII.

An equivalence relation on binary quadratic forms, denoted by $\sim_\lambda$ and called λ-equivalence, can be defined for each value of λ .

Thus $f \sim_\lambda g$ iff $g = fS$ or $g = -fS$ for some element S of $GL_\lambda(2, \mathbb{Z})$. Note $-(\gamma, \delta, \varepsilon) = (-\gamma, -\delta, -\varepsilon)$. 1-equivalence will simply be called equivalence and denoted $\sim$.

It will often be convenient to write $f \sim_\lambda g$ iff $g = \pm fS$, considering the two possibilities at once. Equations 4.3 then become

$$\pm\bar{\gamma} = t^2\gamma + tv\delta + v^2\varepsilon ,$$

$$\pm\bar{\delta} = 2tu\gamma + (tw+uv)\delta + 2vw\varepsilon ,$$

$$\pm\bar{\varepsilon} = u^2\gamma + uw\delta + w^2\varepsilon .$$

The signs on the left-hand side of the equations are then either all positive or all negative. In the same calculations, the condition that S is in $GL(2, \mathbb{Z})$ may be written as $tw - uv = \pm 1$. Thus the occurrence of

$\pm$ may refer to either of two independent choices. Signs are preserved to be consistent with each choice. It is to be hoped that no confusion will arise.

The main theorem of this chapter can now be stated.

THEOREM 4.4. *Two restricted canonical presentations P, Q present isomorphic groups G, H iff $I(G')/G' \cong I(H')/H'$ and $\mathrm{Pf}(P) \sim_\lambda \mathrm{Pf}(Q)$.*

Proof. Let

$$P = \Big\langle g_1, g_2, g_3, g_4, h_1, h_2;\ [g_2, g_1] = h_1^{\alpha_P},\ [g_3, g_1] = h_2^{\beta_P},$$

$$[g_4, g_1] = [g_3, g_2] = \emptyset,\ [g_4, g_2] = h_2^{-\varepsilon_P},$$

$$[g_4, g_3] = h_1^{\gamma_P} h_2^{\delta_P},\ \text{restricted canonical}\Big\rangle$$

and

$$Q = \Big\langle k_1, k_2, k_3, k_4, L_1, L_2;\ [k_2, k_1] = L_1^{\alpha_Q},\ [k_3, k_1] = L_2^{\beta_Q},$$

$$[k_4, k_1] = [k_3, k_2] = \emptyset,\ [k_4, k_2] = L_2^{-\varepsilon_Q},$$

$$[k_4, k_3] = L_1^{\gamma_Q} L_2^{\delta_Q},\ \text{restricted canonical}\Big\rangle$$

present G, H respectively.

$$\mathrm{Pf}(P) = (\gamma_P/\alpha_P)x^2 + (\delta_P/\beta_P)xy + (\varepsilon_P/\beta_P)y^2 ,$$

$$\mathrm{Pf}(Q) = (\gamma_Q/\alpha_Q)x^2 + (\delta_Q/\beta_Q)xy + (\varepsilon_Q/\beta_Q)y^2 .$$

ONLY IF $G \cong H$ implies $I(G')/G' \cong I(H')/H'$ and hence $\alpha_P = \alpha_Q = \alpha$, $\beta_P = \beta_Q = \beta$, and $\lambda = \beta/\alpha$.

From Theorem 3.12, $\mathrm{Pf}(M_Q) = \pm \mathrm{Pf}(M_P) S$ where

$$\mathrm{Pf}(M_P) = \alpha\gamma_P x_1^2 + \alpha\delta_P x_1 x_2 + \beta\varepsilon_P x_2^2 ,$$

and

$$\mathrm{Pf}(M_Q) = \alpha\gamma_Q x_1^2 + \alpha\delta_Q x_1 x_2 + \beta\varepsilon_Q x_2^2 ,$$

and $S = \begin{bmatrix} t & u \\ v & w \end{bmatrix}$ is in $GL(2,\mathbb{Z})$.

Let $\theta : G \to H$ be an isomorphism. By the choice of the action of $GL(2,\mathbb{Z})$, $h_1\theta = L_1^t L_2^u$ and $h_2\theta = L_1^v L_2^w$. Now $h_1^\alpha \in G'$. So $(h_1^\alpha)\theta = L_1^{\alpha t} L_2^{\alpha u} \in H'$.

$L_1^\alpha \in H'$. Thus $L_2^{\alpha u} \in H'$ implying that $\beta | \alpha u$, since $\{L_1^\alpha, L_2^\beta\}$ is a basis for H' . Since $\alpha | \beta$, then $\lambda | u$.

Equations 4.3 become in this context

$$\pm\alpha\gamma_Q = t^2\alpha\gamma_P + tv\alpha\delta_P + v^2\beta\varepsilon_P ,$$

$$\pm\alpha\delta_Q = 2tu\alpha\gamma_P + (tw+uv)\alpha\delta_P + 2vw\beta\varepsilon_P ,$$

$$\pm\beta\varepsilon_Q = u^2\alpha\gamma_P + uw\alpha\delta_P + w^2\beta\varepsilon_P .$$

Writing u as $U\lambda$ and dividing through by α^2, $\alpha\beta$ and β^2 respectively in the three equations above, gives

$$\pm\gamma_Q/\alpha = t^2\gamma_P/\alpha + tv\lambda\delta_P/\beta + v^2\lambda^2\varepsilon_P/\beta ,$$

$$\pm\delta_Q/\beta = 2tU\gamma_P/\alpha + (tw+Uv)\delta_P/\beta + 2vw\lambda\varepsilon_P/\beta ,$$

$$\pm\varepsilon_Q/\beta = U^2\gamma_P/\alpha + Uw\delta_P/\beta + w^2\varepsilon_P/\beta ,$$

which shows that $\mathrm{Pf}(P) = \pm\mathrm{Pf}(Q)\overline{S}$, where $\overline{S} = \begin{bmatrix} t & U \\ v\lambda & w \end{bmatrix}$. Further, $tw - Uv\lambda = tw - uv = \pm 1$, so $\overline{S} \in GL_\lambda(2,\mathbb{Z})$ and $\mathrm{Pf}(P) \sim_\lambda \mathrm{Pf}(Q)$.

IF Suppose $\mathrm{Pf}(P) = -\mathrm{Pf}(Q)$.

Consider the map $\theta : G \to H$ given by

$$\begin{pmatrix} \rho_1 & \rho_2 & \rho_3 & \rho_4 & \sigma_1 & \sigma_2 \\ g_1 & g_2 & g_3 & g_4 & h_1 & h_2 \end{pmatrix}\theta = k_1^{\rho_1} k_2^{\rho_2} k_3^{\rho_3} k_4^{-\rho_4} L_1^{\sigma_1} L_2^{\sigma_2} .$$

It is clear that θ is 1:1 and onto, and easy to check that θ is a homomorphism. Thus P, Q present isomorphic groups.

Let $S = \begin{bmatrix} t & u \\ v\lambda & w \end{bmatrix}$ be an element of $GL_\lambda(2,\mathbb{Z})$ such that $Pf(Q) = \pm Pf(P)S$, where $\lambda = \beta/\alpha$. Note that $\alpha = \alpha_P = \alpha_Q$ and $\beta = \beta_P = \beta_Q$ by assumption.

If $Pf(Q) = -Pf(P)S$, consider the canonical presentation

$$K = \Big\langle k_1, k_2, k_3, k_4, L_1, L_2;\ [k_2, k_1] = L_1^{\alpha},\ [k_3, k_1] = L_2^{\beta},$$

$$[k_4, k_1] = [k_3, k_2] = \emptyset,\ [k_4, k_2] = L_2^{\varepsilon_Q},$$

$$[k_4, k_3] = L_1^{-\gamma_Q} L_2^{-\delta_Q},\ \text{restricted canonical}\Big\rangle .$$

Clearly, $Pf(K) = -Pf(Q) = Pf(P)S$, and from the above discussion P, Q present isomorphic groups iff P and K do.

So assume $Pf(Q) = Pf(P)S$.

Equations 4.3 become

$$\gamma_Q = t^2\gamma_P + tv\delta_P + v^2\lambda\varepsilon_P ,$$

$$\delta_Q = 2tu\lambda\gamma_P + (tw+uv\lambda)\delta_P + 2vw\lambda\varepsilon_P ,$$

$$\varepsilon_Q = u^2\lambda\gamma_P + uw\delta_P + w^2\varepsilon_P .$$

Also, $tw - uv\lambda = \pm 1$.

Let $\eta = \frac{1}{\alpha}\left(\gamma_P tu + \delta_P uv + \varepsilon_P vw\right)$.

Consider the map $\theta : G \to H$ given by

$$\begin{pmatrix} \rho_1 & \rho_2 & \rho_3 & \rho_4 & \sigma_1 & \sigma_2 \\ g_1 & g_2 & g_3 & g_4 & h_1 & h_2 \end{pmatrix}\theta = k_1^{\eta\rho_4 \pm \rho_1} k_2^{\pm(\rho_2 t + \rho_3 v\lambda)} k_3^{\pm(\rho_2 u + \rho_3 w)} k_4^{\rho_4} L_1^{\tau_1} L_2^{\tau_2} ,$$

where

$$\tau_1 = \sigma_1 t + \sigma_2 v \pm \alpha\eta\rho_4(\rho_2 t + \rho_3 v\lambda)$$

and

$$\tau_2 = \sigma_1 u\lambda + \sigma_2 w \pm \beta\eta\rho_4(\rho_2 u + \rho_3 w) .$$

Suppose $g\theta$ is trivial, where

$$g = g_1^{\rho_1} g_2^{\rho_2} g_3^{\rho_3} g_4^{\rho_4} h_1^{\sigma_1} h_2^{\sigma_2} .$$

Considering the exponents of k_4 and k_1 successively gives $\rho_4 = \rho_1 = 0$.

Considering the exponents of k_2 and k_3 gives $\rho_2 t + \rho_3 v\lambda = 0$ and $\rho_2 u + \rho_3 w = 0$, whence $\rho_2 = \rho_3 = 0$ since $tw - uv\lambda = \pm 1$.

Considering the exponents of L_1, L_2 gives $\sigma_1 t + \sigma_2 v = 0$ and $\sigma_1 u\lambda + \sigma_2 w = 0$ implying that $\sigma_1 = \sigma_2 = 0$. Thus g is trivial and θ is injective.

Let g be as above and $h = g_1^{\mu_1} g_2^{\mu_2} g_3^{\mu_3} g_4^{\mu_4} h_1^{\nu_1} h_2^{\nu_2}$. Then

$$gh = g_1^{\rho_1+\mu_1} g_2^{\rho_2+\mu_2} g_3^{\rho_3+\mu_3} g_4^{\rho_4+\mu_4} h_1^{\kappa_1} h_2^{\kappa_2} ,$$

where

$$\kappa_1 = \sigma_1 + \nu_1 + \alpha\rho_2\mu_1 + \gamma_P\rho_4\mu_3 ,$$

and

$$\kappa_2 = \sigma_2 + \nu_2 + \beta\rho_3\mu_1 - \varepsilon_P\rho_4\mu_2 + \delta_P\rho_4\mu_3 .$$

$$(gh)\theta = k_1^{\eta(\rho_4+\mu_4)\pm(\rho_1+\mu_1)} . k_2^{\pm((\rho_2+\mu_2)t+(\rho_2+\mu_3)v\lambda)}$$

$$k_3^{\pm((\rho_2+\mu_2)u+(\rho_3+\mu_3)w)} . k_4^{\rho_4+\mu_4}$$

$$L_1^{\kappa_1 t+\kappa_2 v\pm\alpha\eta(\rho_4+\mu_4)[(\rho_2+\mu_2)t+(\rho_3+\mu_3)v\lambda]}$$

$$L_2^{\kappa_1\lambda u+\kappa_2 w\pm\beta\eta(\rho_4+\mu_4)[(\rho_2+\mu_2)u+(\rho_3+\mu_3)w]}$$

$$(g\theta)(h\theta) = k_1^{\eta\rho_4 \pm \rho_1} k_2^{\pm(\rho_2 t + \rho_3 v\lambda)} k_3^{\pm(\rho_2 u + \rho_3 w)} k_4^{\rho_4}$$

$$L_1^{\tau_1} L_2^{\tau_2} k_1^{\eta\mu_4 \pm \mu_1} k_2^{\pm(\mu_2 t + \mu_3 v\lambda)} k_3^{\pm(\mu_2 u + \mu_3 w)}$$

$$k_4^{\mu_4} L_1^{\omega_1} L_2^{\omega_2} ,$$

where

$$\omega_1 = \nu_1 t + \nu_2 v \pm \alpha\eta\mu_4(\mu_2 t + \mu_3 v\lambda)$$

and

$$\omega_2 = \nu_1 u\lambda + \nu_2 w \pm \beta\eta\mu_4(\mu_2 u + \mu_3 w) .$$

$(g\theta)(h\theta)$ can be collected to give a normal word in H.

The exponents of the k_i's are the same in $(gh)\theta$ and the collected form of $(g\theta)(h\theta)$.

Expanding the exponent of L_1 in $(gh)\theta$ gives

$$t[\sigma_1 + \nu_1 + \alpha\rho_2\mu_1 + \gamma_P\rho_4\mu_3 \pm \alpha\eta(\rho_2 + \mu_2)(\rho_4 + \mu_4)]$$

$$+ v[\sigma_2 + \nu_2 + \beta\rho_3\mu_1 - \varepsilon_P\rho_4\mu_2 + \delta_P\rho_4\mu_3 \pm \alpha\eta\lambda(\rho_3 + \mu_3)(\rho_4 + \mu_4)] .$$

Expanding the exponent of L_1 in the collected form of $(g\theta)(h\theta)$ gives

$$t[\sigma_1 + \nu_1 \pm \alpha\rho_2(\eta\mu_4 \pm \mu_1) \pm \alpha\eta\rho_4\rho_2 \pm \alpha\eta\mu_4\mu_2]$$

$$+ v[\sigma_2 + \nu_2 \pm \alpha\rho_3\lambda(\eta\mu_4 \pm \mu_1) \pm \alpha\eta\rho_4\rho_3\lambda \pm \alpha\eta\mu_4\mu_3\lambda] \pm \gamma_Q\rho_2(\mu_2 u + \mu_3 w) .$$

Many of the terms in the above expressions are identical, remembering $\alpha\lambda = \beta$. For the two expressions to be equal it is sufficient to show that

$$t(\gamma_P\rho_4\mu_3 \pm \alpha\eta\rho_4\mu_2) + v(\delta_P\rho_4\mu_3 - \varepsilon_P\rho_4\mu_2 \pm \beta\eta\rho_4\mu_3) = \pm\gamma_Q\rho_4(\mu_2 u + \mu_3 w) .$$

The left hand side of this proposed equation equals

$$\rho_4\left[t\gamma_P\mu_3 \pm t\mu_2(\gamma_P tu+\delta_P uv+\varepsilon_P vw)+v\delta_P\mu_3-\varepsilon_P\mu_2 v\pm\lambda\mu_3 v(\gamma_P tu+\delta_P uv+\varepsilon_P vw)\right]$$

$$= \rho_4\left[\pm\mu_2 u\left(t^2\gamma_P+\delta_P tv\right)\pm\mu_2\varepsilon_P v(tw\mp 1)\pm\mu_3\left(t\gamma_P(\pm 1+uv\lambda)+v\delta_P(\pm 1+uv\lambda)+\lambda\varepsilon_P v^2 w\right)\right]$$

$$= \pm\rho_4(\mu_2 u+\mu_3 w)\left(t^2\gamma_P+\delta_P tv+\lambda\varepsilon_P v^2\right) \text{ , using } tw - uv\lambda = \pm 1 \text{ in various guises,}$$

$$= \pm\gamma_Q\rho_4(\mu_2 u+\mu_3 w) \text{ , as required.}$$

Similarly, to check whether the exponent of L_2 in $(gh)\theta$ is the same as the exponent of L_2 in the collected form of $(g\theta)(h\theta)$, it is sufficient to check whether

$$u(\gamma_P\rho_4\mu_3\lambda\pm\beta\eta\rho_4\mu_2) + w(\delta_P\rho_4\mu_3-\varepsilon_P\rho_4\mu_2\pm\beta\eta\rho_4\mu_3)$$

$$= \pm\varepsilon_Q\rho_4(\mu_2 t+\mu_3 u\lambda) \pm \delta_Q\rho_4(\mu_2 u+\mu_3 w) .$$

The left hand side of this proposed equation equals

$$\pm\rho_4\mu_2\left[tu^2\lambda\gamma_P+u^2 v\lambda\delta_P+uvw\lambda\varepsilon_P\mp w\varepsilon_P\right]$$

$$\pm\rho_4\mu_3\left[\pm u\lambda\gamma_P\pm w\delta_P+\lambda wtu\gamma_P+\lambda uvw\delta_P+\lambda vw^2\varepsilon_P\right] .$$

The right hand side equals

$$\pm\rho_4\mu_2\left[2tu^2\lambda\gamma_P+(tuw+u^2 v\lambda)\delta_P+2uvw\lambda\varepsilon_P-tu^2\lambda\gamma_P-tuw\delta_P-tw^2\varepsilon_P\right]$$

$$\pm\rho_4\mu_3\left[2tuw\lambda\gamma_P+(tw^2+uvw\lambda)\delta_P+2vw^2\lambda\varepsilon_P-u^2 v\lambda^2\gamma_P-uvw\lambda\delta_P-vw^2\lambda\varepsilon_P\right] ,$$

which are the same since

$$2tuw\lambda - u^2 v\lambda^2 = tuw\lambda + u\lambda(tw-uv\lambda) - tuw\lambda \pm u\lambda$$

and

$$tw^2 - uvw\lambda = \pm w .$$

Thus θ is a homomorphism.

Now

$$g_1\theta = k_1^{\pm 1} , \quad g_4\theta = k_1^{\eta}k_4 ,$$

$$g_2\theta = k_2^{\pm t}k_3^{\pm u} \ , \quad g_3\theta = k_2^{\pm v\lambda}k_3^{\pm w} \ ,$$

$$h_1\theta = L_1^{t}L_2^{u\lambda} \ , \quad h_2\theta = L_1^{v}L_2^{w} \ .$$

Since $tw - uv\lambda = \pm 1$, a set of generators of H are in the image of θ . Thus the map is onto, θ is an isomorphism, and the theorem is proved. □

PROPOSITION 4.5. *The discriminant and order of the Pfaffian of a restricted canonical presentation are invariants of the associated group.*

Proof. By Theorem 4.4 it is enough to show that the discriminant and order of a binary quadratic form are preserved under λ-equivalence.

Let $f = (\gamma, \delta, \varepsilon)$ and $S = \begin{bmatrix} t & u \\ v\lambda & w \end{bmatrix}$ be an element of $GL_\lambda(2, \mathbb{Z})$.

Then

$$\begin{aligned}
\Delta(fS) &= \left(2tu\gamma+(tw+uv\lambda)\delta+2v\lambda w\varepsilon\right)^2 - 4\left(t^2\gamma+tv\lambda\delta+v^2\lambda^2\varepsilon\right)\left(u^2\gamma+uw\delta+w^2\varepsilon\right) \\
&= 4t^2u^2\gamma + (tw+uv\lambda)^2\delta^2 + 4v^2\lambda^2w^2\varepsilon^2 + 4tu\gamma(tw+uv\lambda)\delta + 8tuv\lambda w\gamma\varepsilon \\
&\quad + 4v\lambda w\varepsilon(tw+uv\lambda)\delta - 4t^2u^2\gamma - 4t^2uvw\lambda\gamma\delta - 4t^2w^2\gamma\varepsilon - 4tv\lambda u^2\gamma\delta \\
&\quad - 4uv\lambda tw\delta^2 - 4tv\lambda w^2\delta\varepsilon - 4v^2\lambda^2u^2\gamma\varepsilon - 4uv^2\lambda^2w\delta\varepsilon - 4v^2\lambda^2w^2\varepsilon^2 \\
&= (tw-uv\lambda)^2\delta^2 - 4\gamma\varepsilon\left(t^2w^2-2tuv\lambda w+u^2v^2\lambda^2\right) \\
&= \delta^2 - 4\gamma\varepsilon = \Delta(f) \ .
\end{aligned}$$

By Equations 4.3, $o(fS) \mid o(f)$. By symmetry, $o(f) \mid o(fS)$, and thus $o(f) = o(fS)$. □

The invariants α, β, $\Delta\left(\mathrm{Pf}(P)\right)$, $o\left(\mathrm{Pf}(P)\right)$ do not determine the isomorphism type of the group presented uniquely. The examples of Proposition B of Grunewald and Scharlau (1979), reproduced in section VI, is a pair of nonisomorphic groups with these 4 invariants identical. However I propose the following.

CONJECTURE 4.6. *Two groups have these four invariants equal iff they have the same set of finite quotients.*

III. Reduced Binary Quadratic Forms

Theorem 4.4 essentially reduced the isomorphism problem for groups in $T(4, 2)$ to the problem of determining when two binary quadratic forms are λ-equivalent. There is a wealth of classical material on binary quadratic forms. The literature concentrates on equivalence under the action of $SL(2, \mathbb{Z})$, and as far as I could determine, does not explicitly solve this problem. Thus an account is given here of binary quadratic forms and λ-equivalence, basically adapting the classical ideas and proofs for this context. Most of the notation and terminology used is modelled on that of the literature. Specific references are given as they arise.

Binary quadratic forms are classified into four types, depending on the value of the discriminant.

(i) If $\Delta(f) < 0$, then f is a *definite form.*

(ii) If $\Delta(f) = 0$, then f is a *degenerate zero form.*

(iii) If $\Delta(f) = k^2$, where $k \in \mathbb{Z}^+$, then f is a *zero form.*

(iv) If $\Delta(f) > 0$ and $\Delta(f) \neq k^2$, then f is an *indefinite form.*

This section gives a normal form for the first three types, and criteria which immediately determine when two normal forms are equivalent. The behaviour of indefinite forms is in general more complicated than that of other forms. In particular, there is no easily stated normal form whereby questions of equivalence can be readily answered. Discussion of indefinite forms is thus largely deferred till Section V, where details are needed for the exposition of the algorithm for deciding the λ-equivalence of binary quadratic forms.

This section begins with the following useful equivalences.

LEMMA 4.7. *(i)* $(\gamma, \delta, \varepsilon) \sim (\gamma, -\delta, \varepsilon)$.

(ii) $(\gamma, \delta, \varepsilon) \sim (\varepsilon, \delta, \gamma)$.

(iii) $(\gamma, \delta, \varepsilon) \sim (\gamma, \delta-2\gamma, \varepsilon+\gamma-\delta)$.

(iv) $(\gamma, \delta, \varepsilon) \sim (\gamma+\varepsilon-\delta, \delta-2\varepsilon, \varepsilon)$.

(v) $(\gamma, \delta, \varepsilon) \sim (-\gamma, -\delta, -\varepsilon)$.

Proof. *(i)* Use the substitution $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.

(ii) Use the substitution $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

(iii) Use the substitution $\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$.

(iv) Use the substitution $\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$.

(v) This is a consequence of the definition of the equivalence relation.

A form $(\gamma, \delta, \varepsilon)$ with $0 \leq \delta \leq \gamma \leq |\varepsilon|$ is called *reduced*.

PROPOSITION 4.8. *Every binary quadratic form is equivalent to a reduced form.*

Proof. An algorithm is given to effect the reduction. Let $\gamma x^2 + \delta xy + \varepsilon y^2$ be the form.

1. If the form is reduced, stop.
2. If $\gamma < 0$, apply Lemma 4.7 *(v)*.
3. If $\delta < 0$, apply Lemma 4.7 *(i)*.
4. If $\delta > \gamma$, apply Lemma 4.7 *(iii)*, and go to 3.
5. If $\gamma > |\varepsilon|$, apply Lemma 4.7 *(ii)*, and go to 2.

Observe that the algorithm in the above proof uses the specific substitutions of Lemma 4.7. Thus for any form f , an element S of $GL(2, \mathbb{Z})$ is found such that fS or $-fS$ is reduced. This algorithm is used in the main algorithm of Section V.

PROPOSITION 4.9. *Two equivalent definite reduced forms are identical.*

Proof. This proof is modelled on that of Theorem 56 of Dickson (1939). Let $(\gamma, \delta, \varepsilon)$ and $(\gamma', \delta', \varepsilon')$ be two equivalent reduced forms. By assumption, $\delta^2 - 4\gamma\varepsilon < 0$ and $\gamma > 0$, so $\varepsilon > 0$. Similarly $\gamma' > 0$ and

$\varepsilon' > 0$.

It can be assumed without loss of generality that $\gamma \geq \gamma'$.

Since the forms are equivalent, Equations 4.3 hold, that is there exists $\begin{bmatrix} t & u \\ v & w \end{bmatrix}$ with $tw - uv = \pm 1$ such that

$$\pm\gamma' = t^2\gamma + tv\delta + v^2\varepsilon ,$$

$$\pm\delta' = 2tu\gamma + (tw+uv)\delta + 2vw\varepsilon ,$$

$$\pm\varepsilon' = u^2\gamma + uw\delta + w^2\varepsilon .$$

For any integers t, v the inequality $t^2 + v^2 \geq 2|tv|$ holds.

Since $\gamma x^2 + \delta xy + \varepsilon y^2$ is reduced,

$$\pm\gamma' \geq \gamma t^2 - \gamma|tv| + \gamma v^2$$

$$\geq |tv|\gamma \geq 0 .$$

Thus all the signs on the left-hand side of the equations above are positive. So $\gamma \geq \gamma' \geq |tv|\gamma$, implying $|tv| = 0$ or 1 .

If $|tv| = 1$, $\gamma \geq \gamma' \geq \gamma$ and so $\gamma' = \gamma$.

If $|tv| = 0$, $\gamma \geq \gamma' \geq \gamma t^2 + \gamma v^2 \geq \gamma$ and again $\gamma' = \gamma$, because t, v are not both zero.

Thus $\gamma = \gamma'$.

If $\gamma = \varepsilon$ and $\gamma' = \varepsilon'$, then $\delta = \delta'$ and the forms are identical.

Suppose $\varepsilon > \gamma$. Then $\gamma = \gamma' > \gamma t^2 - \gamma|tv| + \gamma v^2 \geq \gamma|tv|$. So $tv = 0$.

If $t = 0$, $\gamma > \gamma v^2$ which is impossible. So $v = 0$, $tw = \pm 1$ and $t = \pm 1$.

Now $\delta' = \pm 2u\gamma \pm \delta$, where the $\pm$ signs are independent. Since $0 \leq \delta \leq \gamma$ and $0 \leq \delta' \leq \gamma' = \gamma$, either

(i) $u = 0$, $\delta' = \delta$, and hence $\varepsilon = \varepsilon'$;

(ii) $u = 1$, $\delta = \gamma = \delta'$ and $\varepsilon = \varepsilon'$;

(iii) $u = -1$, $\delta = \gamma = \delta'$ and $\varepsilon = \varepsilon'$.

In all cases the forms are identical and the proposition follows. □

PROPOSITION 4.10. *Every degenerate zero form is equivalent to* γx^2 , $\gamma \geq 0$. *For these forms* $\gamma x^2 \sim \gamma' x^2$ *iff* $\gamma = \gamma'$.

Proof. Let $f = (\gamma, \delta, \varepsilon)$ be a reduced, degenerate zero form $\delta^2 - 4\gamma\varepsilon = 0$. So $\delta^2 = 4\gamma\varepsilon \geq 4\delta^2$, implying $\delta = 0$.

Thus $\gamma\varepsilon = 0$ and $\gamma = 0$ if the form is reduced. Apply Lemma 4.8 *(iv)* to change εy^2 to εx^2 , and Lemma 4.8 *(v)* if necessary to ensure $\varepsilon \geq 0$.

Suppose $\gamma x^2 \sim \gamma' x^2$, with $\gamma, \gamma' \geq 0$.

Equations 4.3 become

$$\gamma' = t^2\gamma ,$$

$$0 = 2tu\gamma ,$$

$$0 = u^2\gamma .$$

If $\gamma = 0$, $\gamma' = 0$ and $\gamma = \gamma'$. Otherwise, $u = 0$, $tw = \pm 1$, $t^2 = 1$ and $\gamma' = \gamma$. □

PROPOSITION 4.11. *Any zero form is equivalent to* $\gamma x^2 + \delta xy$, $0 \leq \gamma \leq \delta/2$. *For these forms* $(\gamma, \delta, 0) \sim (\gamma', \delta, 0)$ *iff*

(i) $\gamma = \gamma'$ *or*

(ii) $\gamma\gamma'/d^2 \equiv \pm 1 \pmod{\delta/d}$ *where* $d = (\gamma, \delta) = (\gamma', \delta)$.

Proof. Let $f = \gamma x^2 + \delta xy + \varepsilon y^2$ be a zero form of discriminant k^2 . Let $\eta = (\delta - k)/2$. Note that since $\delta^2 - 4\gamma\varepsilon = k^2$, either δ and k are both even or both odd, and hence η is an integer.

Let $\sigma = (\eta, \varepsilon)$ and $p\eta + q\varepsilon = \sigma$.

Consider $S = \begin{bmatrix} p & -\varepsilon/\sigma \\ q & \eta/\sigma \end{bmatrix}$, which is in $GL(2, \mathbb{Z})$. Then

$fS = \gamma' x^2 \pm kxy$, since

$$(-\varepsilon/\sigma)^2\gamma + (-\varepsilon/\sigma)(\eta/\sigma)\delta + (\eta/\sigma)^2\varepsilon = (\varepsilon/\sigma^2)(\varepsilon\gamma - \delta(\delta-k)/2 + (\delta-k)^2/4)$$

$$= (\varepsilon/4\sigma^2)(4\gamma\varepsilon - 2\delta^2 + 2\delta k + \delta^2 - 2\delta k + k^2)$$

$$= (\varepsilon/4\sigma^2)(k^2 - \delta^2 + 4\gamma\varepsilon) = 0 .$$

Applying Lemma 4.7 *(iv)* as often as necessary, shows that f is equivalent to $\overline{\gamma}x^2 \pm kxy$ where $0 \leq |\overline{\gamma}| \leq k$. Then Lemma 4.7 *(v)* and *(i)* can be applied if necessary, to prove the first half of the proposition.

Suppose $(\gamma, \delta, 0) \sim (\gamma', \delta, 0)$ where $0 \leq \gamma, \gamma' \leq \delta/2$. Equations 4.3 become

$$\pm\gamma' = t^2\gamma + tv\delta , \tag{1}$$

$$\pm\delta = 2tu\gamma + (tw+uv)\delta , \tag{2}$$

$$0 = u^2\gamma + uw\delta . \tag{3}$$

Also

$$tw - uv = \pm 1 . \tag{4}$$

From (1) it follows that $(\gamma, \delta) = (\gamma', \delta) = d$, say. Factorising (3) gives $u(u\gamma+w\delta) = 0$. If $u = 0$, then $tw = \pm 1$, and $\pm\gamma' = \gamma \pm v\delta$ from (1). From the restrictions on γ, γ' either $v = 0$ and $\gamma = \gamma'$, or $-\gamma' = \gamma - \delta$ and $\gamma = \gamma' = \delta/2$. In both cases *(i)* is met.

Suppose $u\gamma + w\delta = 0$. Then $u = \pm\delta/d$ and $w = \mp\gamma/d$, since $(u, w) = 1$ from (4). Substituting in (4) gives $\mp t\gamma/d \mp v\delta/d = \pm 1$, where the $\pm$ signs on either side of the equation are independent. Thus

$$t\gamma + v\delta = \pm d . \tag{5}$$

Multiplying (5) through by t gives $t^2\gamma + tv\delta = \pm td$ and on comparison with (1), $t = \pm\gamma'/d$.

Substituting back in (5) gives $\gamma'\gamma/d + v\delta = \pm d$, that is

$$\gamma'\gamma/d^2 \equiv \pm 1 \pmod{\delta/d} .$$

Conversely, let $\gamma'\gamma/d^2 \equiv \pm 1 \pmod{\delta/d}$, where $d = (\gamma, \delta) = (\gamma', \delta)$. Choose q such that $\gamma'\gamma/d^2 - q\delta/d = \pm 1$. Then $S = \begin{bmatrix} \gamma'/d & \delta/d \\ -q & -\gamma/d \end{bmatrix}$ is in

$GL(2, \mathbb{Z})$. Let $(\gamma, \delta, 0)S = (\bar{\gamma}, \bar{\delta}, \bar{\varepsilon})$.

$$\bar{\gamma} = (\gamma'/d)^2\gamma - (\gamma'/d)q\delta$$

$$= \gamma'\left(\gamma'\gamma/d^2 - q\delta/d\right) = \pm\gamma' ,$$

$$\bar{\delta} = 2\gamma'\delta\gamma/d^2 - \gamma'\gamma\delta/d^2 - q\delta^2/d$$

$$= \delta\left(\gamma'\gamma/d^2 - q\delta/d\right) = \pm\delta ,$$

$$\bar{\delta}^2 - 4\bar{\gamma}\bar{\varepsilon} = \delta^2$$

and so $\bar{\varepsilon} = 0$. Thus $(\gamma', \delta, 0) \sim (\gamma, \delta, 0)$. □

This section is concluded with observations about reduced indefinite quadratic forms.

Let $\gamma x^2 + \delta xy + \varepsilon y^2$ of discriminant $\Delta > 0$ be reduced. Neither γ nor ε can be zero - otherwise the form is a zero form.

Now $\Delta = \delta^2 - 4\gamma\varepsilon \leq |\gamma\varepsilon| - 4\gamma\varepsilon$ and so $\gamma\varepsilon < 0$. Since $\gamma > 0$, then $\varepsilon < 0$.

IV. Automorphs of Binary Quadratic Forms

An element of $GL(2, \mathbb{Z})$ which leaves a binary quadratic form f invariant is called an *automorph* of f . In the context of this work, elements which send f to $-f$ are also important, and will be called *antiautomorphs*. $\mathrm{Autom}(f)$, the set of automorphs and antiautomorphs of f , is clearly a subgroup of $GL(2, \mathbb{Z})$. Some knowledge of these subgroups is needed for the algorithm of the next section, and is developed here.

LEMMA 4.12. *If* $fS = \pm g$ *for two forms* f *and* g , *then* SAS^{-1} *is in* $\mathrm{Autom}(f)$ *iff* A *is in* $\mathrm{Autom}(g)$.

Proof. Let A be in $\mathrm{Autom}(g)$. Then

$$fSAS^{-1} = \pm gAS^{-1} = \pm gS^{-1} = \pm f .$$

Conversely, let SAS^{-1} be in $\mathrm{Autom}(f)$. Then

$$gA = \pm fSA = \pm fSAS^{-1}.S = \pm fS = \pm g \ . \qquad \square$$

This lemma allows us to restrict computation of automorphs and anti-automorphs to the particular forms discussed in the last section. Before considering the four types of forms, Equations 4.3 are rewritten in the current context. Thus if $\begin{bmatrix} t & u \\ v & w \end{bmatrix}$ is in $\mathrm{Autom}((\gamma, \delta, \varepsilon))$, the following equations hold.

EQUATIONS 4.13.
$$\pm\gamma = t^2\gamma + tv\delta + v^2\varepsilon \ , \tag{1}$$
$$\pm\delta = 2tu\gamma + (tw+uv)\delta + 2vw\varepsilon \ , \tag{2}$$
$$\pm\varepsilon = u^2\gamma + uw\delta + w^2\varepsilon \ , \tag{3}$$
$$tw - uv = \pm 1 \ . \tag{4}$$

PROPOSITION 4.14. *Let f be a definite binary quadratic form. Then* $\mathrm{Autom}(f)$ *is a finite group of order at most* 12 .

Proof. This largely follows Theorem 58 of Dickson (1939). Assume the form is reduced. Proceed as in the proof of Proposition 4.9 using Equations 4.13. (Numbered equations refer to these.) Since the form is reduced, considering equation (1),

$$\begin{aligned} \pm\gamma &\geq \gamma t^2 - \gamma|tv| + \gamma v^2 \\ &\geq |tv|\gamma . \end{aligned} \tag{5}$$

Thus there are no antiautomorphs. Also $|tv| = 0$ or 1 . (Note that $\gamma \neq 0$ because the form is definite.)

Suppose $tv = 0$. From (1), $\gamma = \gamma t^2 + \varepsilon v^2$. If $\varepsilon > \gamma$, then $t \neq 0$. Let $t = 0$. So $\varepsilon = \gamma$ and $uv = \pm 1$. Considering (3), $\gamma = \varepsilon = \gamma u^2 + \delta uw + w^2\gamma \geq \gamma|uw|$, as above. So $|uw| = 0$ or 1 .

If $uw = 0$, then $w = 0$ and $uv\delta = \delta$ from (2). Therefore, when $\gamma = \varepsilon$, $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$ are automorphs. If also $\delta = 0$, then $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ are automorphs.

If $|uw| = 1$, then $uw = -1$ from (3), and $u = \pm 1$, $w = \mp 1$.

Considering (2), $\pm v\delta \mp 2v\delta = \delta \Rightarrow \delta = \mp v\delta$. So $\begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$ and $\begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}$ are also automorphs when $\gamma = \varepsilon$, as are $\begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}$ and $\begin{bmatrix} 0 & -1 \\ -1 & 1 \end{bmatrix}$ when the further condition $\delta = 0$ holds.

Let $v = 0$. Considering (3), $\gamma u^2 + \delta uw + \varepsilon = \varepsilon$. So $u(\gamma u + \delta w) = 0$.

If $u = 0$, then $tw\delta = \delta$ from (2). Thus $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$ are always automorphs, and $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ and $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ are automorphs when $\delta = 0$.

If $\gamma u + \delta w = 0$, then $\delta u \pm \delta = 0$ since $tw = \pm 1$ from (4). The form is reduced which restricts the possible solutions for u . Either $u = 0 = \delta$, which is treated above, or $u = \mp 1$ and $\gamma = \delta \neq 0$. In this latter case, consider (2). $\mp 2t\delta \pm t\delta = \delta$ or $\mp t\delta = \delta$ and $t = \mp 1$. Thus $\begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & -1 \\ 0 & 1 \end{bmatrix}$ are automorphs when $\gamma = \delta$.

Note $tw = -1$ for an automorph of this type.

Suppose $|tv| = 1$. Considering (5), the only possibility is $\gamma = \delta = \varepsilon$. From (1), $tv = -1$ and $t = -v = \pm 1$. Rewriting equation (2) gives

$$1 = 2tu + tw - tu - 2tw = t(u - w) .$$

So $u - w = t$.

Rewriting equation (3) gives

$$1 = u^2 + uw + w^2 = (u - w)^2 + 3uw .$$

So $uw = 0$.

If $u = 0$, then $t = \pm 1$, $v = \mp 1$, $w = \mp 1$.

If $w = 0$, then $t = \pm 1$, $v = \mp 1$, $u = \pm 1$.

So $\begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix}$ are automorphs when $\gamma = \delta = \varepsilon$.

Counting the automorphs establishes the proposition. □

Collecting the automorphs in the particular cases gives

COROLLARY 4.15. *Let* $f = (\gamma, \delta, \varepsilon)$ *be a reduced, definite binary quadratic form. Then*

$$\mathrm{Autom}(f) = \pm\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{if } 0 \le \delta < \gamma < \varepsilon$$

$$\pm\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \pm\begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \quad \text{if } 0 < \delta = \gamma < \varepsilon$$

$$\pm\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \pm\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{if } 0 < \delta < \gamma = \varepsilon$$

$$\pm\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \pm\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \pm\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \pm\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{if } 0 = \delta < \gamma = \varepsilon$$

$$\pm\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \pm\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \pm\begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}, \pm\begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}, \pm\begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix}, \pm\begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\text{if } 0 < \delta = \gamma = \varepsilon .$$

PROPOSITION 4.16. $\mathrm{Autom}(\gamma x^2)$, $\gamma > 0$, *equals*

$$\left\langle \pm\begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}, \pm\begin{bmatrix} 1 & 0 \\ k & -1 \end{bmatrix} \right\rangle, \quad k \in \mathbb{Z} .$$

Proof. Equations 4.14 become here

$$t^2\gamma = \pm\gamma , \quad 2tu\gamma = 0 \quad \text{and} \quad u^2\gamma = 0 .$$

By assumption, $\gamma > 0$, so $u = 0$, $tw = \pm 1$ and v takes any value. □

Note that γx^2 has no antiautomorphs.

PROPOSITION 4.17. $\mathrm{Autom}(f)$, *where* f *is a zero form, is a finite group of order at most* 8 .

Proof. By Proposition 4.11 and Lemma 4.12, it is enough to consider

the form $\gamma x^2 + \delta xy$. Clearly $\mathrm{Autom}(n(\gamma x^2+\delta xy)) = \mathrm{Autom}(\gamma x^2+\delta xy)$, so the form can be assumed to be primitive, that is $(\gamma, \delta) = 1$.

Note $\delta \neq 0$ for these zero forms.

Rewriting Equations 4.13 here

$$\pm\gamma = \gamma t^2 + \delta tv \ , \tag{1}$$

$$\pm\delta = 2tu\gamma + (tw+uv)\delta \ , \tag{2}$$

$$0 = \gamma u^2 + \delta uw \ , \tag{3}$$

$$tw - uv = \pm 1 \ . \tag{4}$$

From (3), $u(\gamma u+\delta w) = 0$.

If $u = 0$, then $tw = \pm 1$ from (4).

Considering (1), $\gamma + \delta tv = \pm\gamma$. One solution of this occurs if $v = 0$. Then if $\gamma \neq 0$, only an automorph is possible. If $\gamma = 0$, an antiautomorph is also possible. In the case of an automorph, (2) becomes $tw\delta = \delta$ and $tw = 1$.

So $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$ are always in $\mathrm{Autom}(\gamma x^2+\delta xy)$.

For an antiautomorph, $tw\delta = -\delta$ and $tw = -1$. So $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ and $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ are in $\mathrm{Autom}(\delta xy)$.

The other solution of $\gamma + \delta tv = \pm v$ is $v = \mp t$ when $\gamma = \delta/2$ and $\begin{bmatrix} t & u \\ v & w \end{bmatrix}$ is an antiautomorph. $\begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix}$ and $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ are then in $\mathrm{Autom}((\delta/2)x^2+\delta xy)$.

Suppose $\gamma u + \delta w = 0$. If $\gamma = 0$, then $\delta w = 0$ and $w = 0$. From (1), $\delta tv = 0$ and $t = 0$. From (2), $uv\delta = \pm\delta$. This case, then gives two automorphs, $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$, and two antiautomorphs, $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

and $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

If $\gamma \neq 0$, then $u = \pm\delta$ and $w = \mp\gamma$, since the form is primitive and $(u, w) = 1$ from (4). Consider automorphs and antiautomorphs separately. For automorphs, (2) becomes

$$\pm 2t\gamma\delta \mp t\gamma\delta \pm v\delta^2 = \delta \Rightarrow \pm t\gamma \pm v\delta = 1 \quad \text{or} \quad t\gamma + v\delta = \pm 1 .$$

Multiplying though by t gives $t^2\gamma + tv\delta = \pm t$. Then $t = \pm\gamma$ on comparison with (1).

Solving for v gives $v = \pm(1-\gamma^2)/\delta$. Thus if $\delta | (1-\gamma^2)$, then

$$\begin{bmatrix} \gamma & \delta \\ (1-\gamma^2)/\delta & -\gamma \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -\gamma & -\delta \\ (\gamma^2-1)/\delta & \gamma \end{bmatrix}$$

are automorphs.

For antiautomorphs, (2) becomes

$$\pm 2t\gamma\delta \mp t\gamma\delta \pm v\delta^2 = -\delta .$$

Solving for t and v gives

$$t = \pm\gamma \quad \text{and} \quad v = \pm(1+\gamma^2)/\delta .$$

Thus if $\delta | (1+\gamma^2)$, then

$$\begin{bmatrix} \gamma & \delta \\ (1+\gamma^2)/\delta & -\gamma \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -\gamma & -\delta \\ -(1+\gamma^2)/\delta & \gamma \end{bmatrix}$$

are antiautomorphs.

A count of the automorphs establishes the proposition. □

Collecting the results gives

COROLLARY 4.18.

$$\mathrm{Autom}(\gamma x^2+\delta xy) = \pm\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \qquad 0 < \gamma < \delta/2,\ \delta \nmid 1-\gamma^2,\ \delta \nmid 1+\gamma^2$$

$$\pm\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \pm\begin{bmatrix}\gamma & \delta\\ (1-\gamma^2)/\delta & -\gamma\end{bmatrix}, \qquad 0 < \gamma < \delta/2,\ \delta \mid 1-\gamma^2,\ \delta \nmid 1+\gamma^2$$

$$\pm\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \pm\begin{bmatrix}\gamma & \delta\\ (1+\gamma^2)/\delta & -\gamma\end{bmatrix}, \qquad 0 < \gamma < \delta/2,\ \delta \nmid 1-\gamma^2,\ \delta \mid 1+\gamma^2$$

$$\pm\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \pm\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, \pm\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}, \pm\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}, \qquad 0 = \gamma < \delta$$

$$\pm\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \pm\begin{bmatrix}-1 & 0\\ 1 & 1\end{bmatrix}, \pm\begin{bmatrix}1 & 2\\ 0 & -1\end{bmatrix}, \pm\begin{bmatrix}1 & 2\\ -1 & -1\end{bmatrix}, \qquad 0 < \gamma = \delta/2\ .$$

PROPOSITION 4.19. *Let* $f = (\gamma, \delta, \varepsilon)$ *be a reduced indefinite binary quadratic form. Then every element of* $\mathrm{Autom}(f)$ *can be written as*

$$\begin{bmatrix}(p-\delta q)/2 & -\varepsilon q\\ \gamma q & (p+\delta q)/2\end{bmatrix},$$ *where* p, q *are integers satisfying*

$$p^2 - \Delta(f)q^2 = \pm 4\ ,$$

or as

$$\begin{bmatrix}s & (r+\delta s)/2\gamma\\ (r-\delta s)/2\varepsilon & -s\end{bmatrix},$$ *where* r, s *are integers satisfying*

$$r^2 - \Delta(f)s^2 = \pm 4\gamma\varepsilon\ .$$

Conversely, every integer solution to the equations $x^2 - \Delta(f)y^2 = \pm 4$, *and* $x^2 - \Delta(f)y^2 = \pm 4\gamma\varepsilon$ *gives an element of* $\mathrm{Autom}(f)$ *of the respective type, when the matrix entries are integers.*

Proof. Let $f = (\gamma, \delta, \varepsilon)$ be a reduced indefinite binary quadratic form. As in the proof of Proposition 4.17, assume that f is primitive. Numbered equations (1)-(4) refer to Equations 4.13.

Let $S = \begin{bmatrix}t & u\\ v & w\end{bmatrix}$ and $\Delta = \Delta(f)$. Equations (1) and (2) can be usefully

factorised as follows:

$$\pm\gamma = \gamma\left(t+v(\delta-\sqrt{\Delta})/2\gamma\right)\left(t+v(\delta+\sqrt{\Delta})/2\gamma\right) , \tag{5}$$

$$\pm\delta = 2\gamma\left(t+v(\delta+\sqrt{\Delta})/2\gamma\right)\left(u+w(\delta-\sqrt{\Delta})/2\gamma\right) \pm \sqrt{\Delta} , \tag{6}$$

$$= 2\gamma\left(t+v(\delta-\sqrt{\Delta})/2\gamma\right)\left(u+w(\delta+\sqrt{\Delta})/2\gamma\right) \mp \sqrt{\Delta} . \tag{7}$$

The signs of the right hand of (6) and (7) depend on the sign of the determinant of S .

Let $\theta = (\delta-\sqrt{\Delta})/2\gamma$, $\psi = (\delta+\sqrt{\Delta})/2\gamma$. Then θ, ψ are irrational and satisfy

$$\gamma x^2 - \delta x + \varepsilon = 0 . \tag{8}$$

Note that $\theta \neq \psi$.

If S is an automorph of determinant 1 , or an antiautomorph of determinant -1 , expand θ using (5) and (6). That is

$$\theta = \frac{2\gamma\left(t+v(\delta+\sqrt{\Delta})/2\gamma\right)\left(u+w(\delta-\sqrt{\Delta})/2\gamma\right)}{2\gamma\left(t+v(\delta-\sqrt{\Delta})/2\gamma\right)\left(t+v(\delta+\sqrt{\Delta})/2\gamma\right)} .$$

More compactly, $\theta = (u+w\theta)/(t+v\theta)$.

For the same two cases, ψ can be expanded using (5) and (7) to give

$$\psi = (u+w\psi)/(t+v\psi) .$$

Then

$$\theta(t+v\theta) = u + w\theta$$

or

$$v\theta^2 + (t-w)\theta - u = 0 .$$

Similarly

$$v\psi^2 + (t-w)\psi - u = 0 .$$

Thus θ, ψ satisfy $vx^2 + (t-w)x - u = 0$. Comparing with equation (8),

$$v = \gamma q , \quad t - w = -\delta q \quad \text{and} \quad u = -\varepsilon q \quad \text{for} \quad q \in \mathbb{Q} .$$

Since the form is primitive, $q \in \mathbb{Z}$.

Let $p = t + w$. Then

$$tw = \pm 1 + uv = \pm 1 - \gamma\varepsilon q^2 ,$$

$$p^2 = (t-w)^2 + 4tw = \delta^2 q^2 \pm 4 - 4\gamma\varepsilon q^2$$

$$= \Delta q^2 \pm 4 .$$

Thus

$$t = \tfrac{1}{2}(p-\delta q) , \quad w = \tfrac{1}{2}(p+\delta q) ,$$

and

$$S = \begin{bmatrix} \tfrac{1}{2}(p-\delta q) & -\varepsilon q \\ \gamma q & \tfrac{1}{2}(p+\delta q) \end{bmatrix}$$

where $p^2 - \Delta q^2 = \pm 4$.

If S is an antiautomorph of determinant 1 , or an automorph of determinant -1 , expand θ using (5) and (7), and ψ using (5) and (6) to give

$$\theta = (u+w\psi)/(t+v\psi) \quad \text{and} \quad \psi = (u+w\theta)/(t+v\theta) .$$

Thus

$$\theta(t+v\psi) = u + w\psi ,$$

and

$$\psi(t+v\theta) = u + w\theta .$$

Subtracting these two equations gives

$$t(\theta-\psi) = w(\psi-\theta) \quad \text{and} \quad t = -w .$$

Substituting in (2) gives $2tu\gamma - (t^2-uv)\delta - 2vt\varepsilon = \mp\delta$. Substituting in the determinant (4) gives $t^2 + uv = \mp 1$. (In both cases the upper signs correspond to $\det S = 1$.) Combining these two substitutions and simplifying gives

$$tu\gamma - t^2\delta - tv\varepsilon = 0 .$$

If $t = 0$, then $v^2\varepsilon = \mp\gamma$ from (1). The only solution for this occurs when $\gamma = -\varepsilon$ and S is an antiautomorph. Note that $\varepsilon < 0$ and $\gamma > 0$ since the form is reduced.

Then $uv = -1$. So $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ are in

$\mathrm{Autom}(\gamma x^2+\delta xy-\gamma y^2)$. These correspond to the solutions $r = \pm 2\gamma$, $s = 0$ of $r^2 - \Delta s^2 = -4\gamma\varepsilon = 4\gamma^2$. Otherwise

$$u\gamma = t\delta + v\varepsilon . \tag{9}$$

Consider (1) as a quadratic equation in v . Solving $t^2\gamma + tv\delta + v^2\varepsilon = \mp\gamma$ gives

$$\begin{aligned} w &= \left(-t\delta \pm \sqrt{t^2\delta^2 - 4\gamma\varepsilon\left(t^2 \pm 1\right)}\right)/2\varepsilon \\ &= \left(-t\varepsilon \pm \sqrt{t^2\Delta \mp 4\gamma\varepsilon}\right)/2\varepsilon . \end{aligned}$$

For v to be rational, $t^2\Delta \mp 4\gamma\varepsilon$ must be a perfect square.

Let $r^2 = t^2\Delta \mp 4\gamma\varepsilon$, and let $s = t$. Then $v = (-s\delta \pm r)/2\varepsilon$. Solving (9) for u gives $u = (s\delta \pm r)/2\gamma$. Choosing signs as necessary

$$S = \begin{bmatrix} s & (r+s\delta)/2\gamma \\ (r-s\delta)/2\varepsilon & -s \end{bmatrix} ,$$

where r, s are integers satisfying $r^2 - \Delta s^2 = \pm 4\gamma\varepsilon$.

The converse of the proposition is proved by routinely checking that Equations 4.13 hold for the matrices chosen. Some sample calculations are given.

Let

$$S = \begin{bmatrix} (p-\delta q)/2 & -\varepsilon q \\ \gamma q & (p+\delta q)/2 \end{bmatrix} , \text{ where } p^2 - \Delta q^2 = \pm 4 .$$

Then

$$\begin{aligned} \det S &= (p-\delta q)(p+\delta q)/4 + \gamma\varepsilon q^2 \\ &= \tfrac{1}{4}\left(p^2 - \delta^2 q^2 + 4\gamma\varepsilon q^2\right) \\ &= \pm 1 . \end{aligned}$$

Verifying (1) of Equations 4.13,

$$\gamma' = \tfrac{1}{4}(p-\delta q)^2\gamma + (\gamma q/2)(p-\delta q)\delta + \gamma^2 q^2\varepsilon$$

$$= (\gamma/4)\{p^2-2\delta pq+\delta^2q^2+2\delta pq-2\delta^2q^2+4\gamma\varepsilon q^2\}$$

$$= (\gamma/4)\{p^2-\Delta q^2\} = \pm\gamma \ .$$

Verifying (2),

$$\delta' = -\varepsilon q(p-\delta q)\gamma + \tfrac{1}{4}(p-\delta q)(p+\delta q)\delta - \gamma\varepsilon q^2\delta + \gamma q(p \times \delta q)\varepsilon$$

$$= (\delta/4)\{4\gamma\varepsilon q^2+p^2-\delta^2q^2-4\gamma\varepsilon q^2+4\gamma\varepsilon q^2\}$$

$$= (\delta/4)\{p^2-\Delta q^2\} = \pm\delta \ .$$

Verifying (1) for

$$S = \begin{bmatrix} s & (r+s\delta)/2\gamma \\ (r-s\delta)/2\varepsilon & -s \end{bmatrix}$$

where $r^2 - \Delta s^2 = \pm 4\gamma\varepsilon$,

$$s^2\gamma + \delta s(r-s\delta)/2\varepsilon + \varepsilon(r-s\delta)^2/4\varepsilon^2 = (\gamma/4\gamma\varepsilon)\{4\gamma\varepsilon s^2+2rs\delta-2s^2\delta^2+r^2-2rs\delta+s^2\delta^2\}$$

$$= (\gamma/4\gamma\varepsilon)\{r^2-\Delta s^2\} = \pm\gamma \ .$$

The other verifications are omitted. □

Let $t = (p-\delta q)/2$ and $w = (p+\delta q)/2$ where $p + \sqrt{\Delta}q$ is a solution of $x^2 - \Delta y^2 = \pm 4$ and $\Delta' = \delta^2 - 4\gamma\varepsilon$.

Then $2t + 2w = 2p$, and so $2t$ and $2w$ are both even integers or both odd.

Now $tw = \{p^2-\delta^2q^2\}/4 = \{p^2-\Delta q^2\}/4 + \gamma\varepsilon q^2 = \pm 1 + \gamma\varepsilon q^2$, which is an integer. So $2t$ and $2w$ are both even, and t and w are integers.

Thus every solution of $x^2 - \Delta(f)y^2 = \pm 4$ leads to an element of Autom(f) .

This may not be true for solutions of $x^2 - \Delta y^2 = \pm 4\gamma\varepsilon$. A discussion of this is deferred to the next section.

V. λ-Equivalence of Binary Quadratic Forms

This section describes an algorithm for determining whether two binary quadratic forms are λ-equivalent. The algorithm can be used as a tool for investigating groups in $\Gamma(4, 2)$. Some examples of its use are given in the next section.

We begin with a useful lemma.

LEMMA 4.20. *Let f, g be two binary quadratic forms such that $fS = \pm gT$. Then $f \sim_\lambda g$ iff SAT^{-1} is in $GL_\lambda(2, \mathbb{Z})$ for some element A of $\mathrm{Autom}(fS)$.*

Proof. If $f \sim_\lambda g$, then $fU = \pm g$ for some U in $GL_\lambda(2, \mathbb{Z})$. Then $fSS^{-1}UT = fUT = \pm gT = \pm fS$. So $S^{-1}UT = A$ for some A in $\mathrm{Autom}(fS)$ or $U = SAT^{-1}$.

If A is in $\mathrm{Autom}(fS)$, then $fSAT^{-1} = \pm fST^{-1} = \pm g$. Since SAT^{-1} is in $GL_\lambda(2, \mathbb{Z})$, $f \sim_\lambda g$.

THEOREM 4.21. *There is an algorithm to decide whether two binary quadratic forms are λ-equivalent.*

Proof. Let f, g be two binary quadratic forms. The steps of the algorithm are as follows. // denotes a possible termination point of the algorithm.

1. Compute $\Delta(f)$ and $\Delta(g)$.
2. If $\Delta(f) \neq \Delta(g)$, then the two forms are not λ-equivalent by Proposition 4.5. //

   Otherwise, let $\Delta = \Delta(f) = \Delta(g)$.
3. If $o(f) \neq o(g)$, then the two forms are not λ-equivalent by Proposition 4.5. //

The algorithm proceeds differently for each of the four types of forms. Each case is considered separately starting from step 4 of the algorithm.

(i) $\Delta > 0$

4. Calculate S, T in $GL(2,\mathbb{Z})$ such that $f' = fS$ or $-fS$ is reduced, and $g' = gT$ or $-gT$ is reduced.

   The algorithm of Proposition 4.8 is used for this step.

5. If $f' \neq g'$, then f and g are not equivalent, and hence not λ-equivalent, by Proposition 4.9. //

6. Calculate SAT^{-1} for the finite number (≤ 12) of elements A in $\mathrm{Autom}(fS)$.

   Corollary 4.15 lists the elements of $\mathrm{Autom}(fS)$.

7. SA_0T^{-1} is in $GL_\lambda(2,\mathbb{Z})$, with A_0 in $\mathrm{Autom}(fS)$, then $fSA_0T^{-1} = \pm g$ and $f \sim_\lambda g$, by Lemma 4.20. //

8. If no SAT^{-1} is in $GL_\lambda(2,\mathbb{Z})$, where A runs over all elements of $\mathrm{Autom}(fS)$, then $f \not\sim_\lambda g$ by Lemma 4.20. //

   Thus the procedure terminates for definite forms.

(ii) $\Delta = 0$

4. If f and g are the trivial degenerate zero form, then $f \sim_\lambda g$. //

5. If f is the trivial degenerate zero form, and g is not, or *vice versa*, then $f \not\sim_\lambda g$. //

6. Calculate S, T in $GL(2,\mathbb{Z})$ such that $fS = \pm\gamma x^2$ and $gT = \pm\gamma' x^2$, where γ and $\gamma' > 0$.

   Propositions 4.8 and 4.10 are used for this step.

7. If $\gamma \neq \gamma'$, then $f \not\sim_\lambda g$ by Proposition 4.10. //

Some comments are needed before step 8 is specified. When $\gamma = \gamma'$, then $fS = \pm gT$, and by Lemma 4.20, $f \sim_\lambda g$ iff SAT^{-1} is in $GL_\lambda(2,\mathbb{Z})$ for some A in $\mathrm{Autom}(\gamma x^2)$. Let $S = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $T^{-1} = \begin{bmatrix} t & u \\ v & w \end{bmatrix}$.

Using Proposition 4.16, and noting that U is in $GL_\lambda(2, \mathbb{Z})$ iff $-U$ is, it must be decided whether $\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}\begin{bmatrix} t & u \\ v & w \end{bmatrix}$ or $\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 1 & 0 \\ k & -1 \end{bmatrix}\begin{bmatrix} t & u \\ v & w \end{bmatrix}$ is in $GL_\lambda(2, \mathbb{Z})$ for some integer k .

This is true if there is an integer k satisfying $dtk \equiv -dv - ct \pmod{\lambda}$ or $dtk \equiv dv - ct \pmod{\lambda}$.

There is an integer k satisfying these congruences iff $(dt, \lambda) | dv-ct$ or $(dt, \lambda) | dv+ct$.

8. Calculate $Z = \left(S(2, 2)T^{-1}(1, 1), \lambda\right)$.

9. If $Z \left| \left(S(2, 2)T^{-1}(2, 1)-S(2, 1)T^{-1}(1, 1)\right)\right.$, then $f \sim_\lambda g$ by the above discussion. //

10. Similarly, if $Z \left| \left(S(2, 2)T^{-1}(2, 1)+S(2, 1)T^{-1}(1, 1)\right)\right.$, then $f \sim_\lambda g$ .//

11. Otherwise, $f \not\sim_\lambda g$. //

If the algorithm terminates at step 9 or 10, an element U in $GL_\lambda(2, \mathbb{Z})$ is readily calculated, such that $fU = \pm g$.

(iii) $\Delta = k^2$

4. Calculate S, T in $GL(2, \mathbb{Z})$ such that $\pm fS = \gamma x^2 + kxy$ and $\pm gT = \gamma' x^2 + kxy$, with $0 \le \gamma, \gamma' \le \delta/2$.

Proposition 4.11 is used for this step.

5. Test whether $\gamma x^2 + kxy \sim \gamma' x^2 + kxy$ using the criteria given in Proposition 4.11.

If these two forms are inequivalent, $f \not\sim_\lambda g$. //

6. Calculate U such that $fSU = \pm gT$, again using Proposition 4.11.

7. Calculate $SUAT^{-1}$ for the finite number (≤ 8) of elements A in $\mathrm{Autom}(fS)$.

Corollary 4.18 lists the elements of Autom(fS) .

8. If SUA_0T^{-1} is in $GL_\lambda(2,\mathbb{Z})$, with A_0 in Autom(fS) , then $fSUA_0T^{-1} = \pm g$, and $f \sim_\lambda g$. //

9. If no SUA_0T^{-1} is in $GL_\lambda(2,\mathbb{Z})$, where A runs over all elements of Autom(fS) , then $f \sim_\lambda g$ by Lemma 4.20. //

(iv) $\Delta > 0$, $\Delta \neq k^2$

4. Calculate S, T In $GL(2,\mathbb{Z})$ such that $f' = fS$ or $-fS$ is reduced, and $g' = gT$ or $-gT$ is reduced.

The description of the algorithm is interrupted here in order to develop some theory about the equivalence of reduced indefinite forms. The treatment given here is not the only one possible. For example, the theory in Sections 43-46 of Dickson (1939) could be adapted to give a different algorithm to determine when two indefinite forms are equivalent. The algorithm actually described, largely based on Section 48 of Jones (1950), is chosen for compatibility with the discussion of reduced forms in the other three cases, and for self-containment of the relevant theory. Questions of which method might be preferable for practical implementation are not considered.

Let $f' = (\gamma, \delta, \varepsilon)$ and $g' = (\gamma', \delta', \varepsilon')$. Assume without loss of generality that $\gamma' \leq \gamma$. Also assume that f' and g' are primitive since $f \sim g$ iff $(\gamma/o(f), \delta/o(f), \varepsilon/o(f)) \sim (\gamma'/o(f), \delta'/o(f), \varepsilon'/o(f))$. Note that $0 \leq \delta \leq \gamma \leq -\varepsilon$, $0 \leq \delta' \leq \gamma' \leq -\varepsilon'$, and $\delta, \delta' < \sqrt{\Delta}$, since f' and g' are reduced.

Suppose that $f'U = \pm g'$, where $U = \begin{bmatrix} t & u \\ v & w \end{bmatrix}$ and $t/v > 0$. Then

$$\pm\gamma' = t^2\gamma + tv\delta + v^2\varepsilon \qquad (1)$$

$$= \gamma\left(t - v(\sqrt{\Delta}-\delta)/2\gamma\right)\left(t + v(\sqrt{\Delta}+\delta)/2\gamma\right) .$$

Let $\theta = (\sqrt{\Delta}-\delta)/2\gamma$. If $f'U = g'$, divide through by γv^2 to give

$$(t/v - \theta)\left(t/v + (\sqrt{\Delta}+\delta)/2\gamma\right) = \gamma'/\gamma v^2 > 0 .$$

By supposition, the second of the terms in the product above is positive, and hence so is the first, that is $t/v > \theta$. So

$$t/v + (\sqrt{\Delta}+\delta)/2\gamma > \sqrt{\Delta}/\gamma > 2\sqrt{\gamma}.\sqrt{-\varepsilon}/\gamma > 2 .$$

So

$$t/v - \theta < \gamma'/2\gamma v^2 \leq 1/2v^2 .$$

By Lemma 1.7, t/v is a convergent in the continued fraction expansion of θ .

If $f'U = -g'$, divide (1) through by εt^2 and factorise to give

$$0 < -\gamma'/\varepsilon t^2 = (v/t - 1/\theta)(v/t + 2\gamma/\sqrt{\Delta}+\delta) .$$

Reasoning as before, $v/t > 1/\theta$ and

$$v/t + 2\gamma/\sqrt{\Delta}+\delta > \left(2\gamma(\sqrt{\Delta}+\delta)+2\gamma(\sqrt{\Delta}-\delta)\right)/\Delta-\delta^2 = -\sqrt{\Delta}/\varepsilon .$$

Then

$$v/t - 1/\theta < \varepsilon/\sqrt{\Delta} \; . \; \gamma'/\varepsilon t^2 < 1/2t^2 .$$

By Lemma 1.7, v/t is a convergent in the continued fraction expansion of $1/\theta$. By Lemma 1.5, t/v is a convergent in the expansion of θ .

Let t/v be the nth convergent in the expansion of θ . By Lemma 1.6,

$$\theta = (t\psi+r)/(v\psi+s)$$

where r/s is the $(n-1)$th convergent and ψ is the $(n+1)$th complete quotient. By properties of convergents, $ts - rv = \pm 1$. By Lemma 1.8, θ has a periodic continued fraction with a period of length L say, starting from the kth term.

Suppose $n > k + L$.

Then ψ is also the $(n-L+1)$th complete quotient, and $\theta = (\bar{t}\psi+\bar{r})/(\bar{v}\psi+\bar{s})$, by Lemma 1.6, where $\bar{t}/\bar{v}$ and $\bar{r}/\bar{s}$ are the $(n-L)$th and $(n-L-1)$th convergents respectively. Again, $\bar{t}\bar{s} - \bar{r}\bar{v} = \pm 1$. Then $\psi = (-\bar{s}\theta+\bar{r})/(\bar{v}\theta-\bar{t})$ and

$$\theta = [t(-\bar{s}\theta+\bar{r})/(\bar{v}\theta-\bar{t})+r]/[v(-\bar{s}\theta+\bar{r})/(\bar{v}\theta-\bar{t})+s]$$

$$= (a\theta+b)/(c\theta+d) , \qquad (2)$$

where

$$a = r\bar{v} - t\bar{s} , \quad b = t\bar{r} - \bar{t}r ,$$

$$c = s\bar{v} - \bar{s}v , \quad d = v\bar{r} - s\bar{t}$$

$$ad - bc = (ts-rv)(\bar{t}\bar{s}-\bar{r}\bar{v}) = \pm 1 .$$

Multiplying (2) out,

$$c\theta^2 + (d-a)\theta - v = 0$$

or

$$c(\sqrt{\Delta}-\delta)^2 + 2\gamma(d-a)(\sqrt{\Delta}-\delta) - 4\gamma^2 b = 0 .$$

Considering the coefficient of $\sqrt{\Delta}$,

$$-2c\delta + 2\gamma(d-a) = 0 . \qquad (3)$$

Also

$$c\Delta + c\delta^2 - 2\gamma(d-a)\delta - 4\gamma^2 b = 0 .$$

So

$$-c.4\gamma\varepsilon + 2c\delta^2 - 2\gamma(d-a)d - 4\gamma^2 b = 0 ,$$

implying

$$-c\varepsilon - \gamma b = 0 . \qquad (4)$$

Now $c = \gamma q$, for some $q \in \mathbb{Q}$. Then $d - a = \delta q$ from (3), and $b = -\varepsilon q$ from (4).

Since f' is primitive, q is an integer.

Continuing as in the proof of Proposition 4.19,

$$B = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} (p-\delta q)/2 & -\varepsilon q \\ \gamma q & (p+\delta q)/2 \end{bmatrix} \text{ where } p^2 - \Delta q^2 = 4 .$$

Then B is in Autom(f') by Proposition 4.19. Also B^{-1} is in Autom(f') and $f'B^{-1}U = \pm f'U = \pm g$.

$$B^{-1}U = \begin{bmatrix} \pm d & \mp b \\ \mp c & \pm a \end{bmatrix} \begin{bmatrix} t & * \\ v & * \end{bmatrix} = \begin{bmatrix} t' & * \\ v' & * \end{bmatrix} ,$$

where

$$\begin{aligned} t' &= \pm(v\bar{r}-s\bar{t})t \mp (t\bar{r}-\bar{t}r)v \\ &= \pm\bar{t}(rv-st) = \mp\bar{t} , \\ v' &= \mp(s\bar{v}-\bar{s}v)t \pm (r\bar{v}-t\bar{s})v \\ &= \pm\bar{v}(rt-st) = \mp\bar{v} . \end{aligned}$$

The following has been established.

If there is an element $U = \begin{bmatrix} t & * \\ v & * \end{bmatrix}$ in GL$(2,\mathbb{Z})$, with $t/v > 0$, such that $f'U = \pm g'$, then there is an element $\bar{U} = \begin{bmatrix} \bar{t} & * \\ \bar{v} & * \end{bmatrix}$ such that $f'\bar{U} = \pm g'$ and $\bar{t}/\bar{v}$ is a convergent in the expansion of $\theta = (\sqrt{\Delta}-\delta)/2\gamma$. Further, if the expansion of θ has period of length L , starting from the kth term, then $\bar{U}$ can be chosen so that $\bar{t}/\bar{v}$ is one of the first $k + L$ convergents.

It is now shown that if $f'V = \pm g'$, then there is an element $U = \begin{bmatrix} t & * \\ v & * \end{bmatrix}$ of GL$(2,\mathbb{Z})$, with $t/v > 0$ such that $f'U = \pm g'$.

Let $V = \begin{bmatrix} \bar{t} & \bar{v} \\ \bar{u} & \bar{w} \end{bmatrix}$ in GL$(2,\mathbb{Z})$ take f' to g' . By Proposition 4.19,

$$P = \begin{bmatrix} (p-\delta q)/2 & -\varepsilon q \\ \gamma q & (p+\delta q)/2 \end{bmatrix} , \text{ where } p^2 - \Delta q^2 = 4 ,$$

is an automorph of f' .

Let $PV = \begin{bmatrix} t & * \\ v & * \end{bmatrix}$. Then

$$t = \bar{t}(p-\delta q)/2 - \bar{v}\varepsilon q$$

and

$$v = \bar{t}\gamma q - \bar{v}(p+\delta q)/2 .$$

Then

$$tv = \bar{t}^2\gamma q(p-\delta q)/2 - \bar{v}^2\varepsilon q(p+\delta q)/2 - \gamma\varepsilon q^2\bar{t}\bar{v} + \bar{t}\bar{v}\left(p^2-\delta^2q^2\right)/4$$

$$= \bar{t}^2\gamma q(p-\delta q)/2 - \bar{v}^2\varepsilon q(p+\delta q)/2 + \bar{t}\bar{v} - 2\gamma\varepsilon q^2 . \qquad (5)$$

Choose $p + \sqrt{\Delta}q$ to be a large positive solution of $p^2 - \Delta q^2 = 4$. Such a solution exists by Proposition 1.13. Then $(p-\sqrt{\Delta}q)(p+\sqrt{\Delta}q) = 4 > 0$, and both terms in the product are positive.

So $0 < p - \sqrt{\Delta}q < p - \delta q < p + \delta q$ and $\varepsilon < 0$.

All terms in (5) are positive, except possibly $\bar{t}\bar{v}$. Thus tv can be made positive by choosing a sufficiently large q. Clearly t/v is then positive.

The description of the algorithm is now resumed.

5. Calculate the first $k + L$ convergents in the continued fraction expansion of $\theta = (\sqrt{\Delta}-\delta)/2\gamma$, where the expansion has period of length L starting from the kth term.

   Let t_i/v_i denote the ith convergent.

6. If $t_i^2\gamma + t_iv_i\delta + v_i^2\varepsilon \neq \pm\gamma'$ for $1 \leq i \leq k + L$, then $f' \not\sim g'$ by the above discussion. //

7. Let $t_i^2\gamma + t_iv_i\delta + v_i^2\varepsilon = \pm\gamma'$. Solve the equations

$$t_iw - v_iu = \pm 1 ,$$

$$2t_iu\gamma + \left(t_iw+v_iu\right)\delta + 2v_iw\varepsilon = \pm\delta' ,$$

$$u^2\gamma + uw\delta + w^2\varepsilon = \pm\varepsilon' ,$$

for u, w to find a U such that $f'U = \pm g'$.

The next step involves considering whether $SUAT^{-1}$ is in $GL_\lambda(2, \mathbb{Z})$ for some element A of Autom(fS) . By Proposition 4.19, there are possibly four types of elements of Autom(fS) connected to solutions of the equations $x^2 - \Delta y^2 = \pm 4$ and $x^2 - \Delta y^2 = \pm 4\gamma'\varepsilon'$. First, the automorphs connected with the solution of $x^2 - \Delta y^2 = 4$ are considered.

By Proposition 1.13, all solutions of this equation can be written as

$$p + \sqrt{\Delta}q = \pm 2\left[\left(p_1 + \sqrt{\Delta}q_1\right)/2\right]^n ,$$

where $p_1 + \sqrt{\Delta}q_1$ is the minimal positive solution.

8. Compute the minimal positive solution $p_1 + \sqrt{\Delta}q_1$ of $p^2 - \Delta q^2 = 4$.

Note that $\Delta \equiv 0$ or $1 \pmod 4$.

If $\Delta = 4\overline{\Delta}$ and $p^2 - \Delta q^2 = 4$, then p is even. Thus in this case, the minimal positive solution is obtained from the fundamental solution of $p^2 - \overline{\Delta}q^2 = 1$. Suppose $\Delta \equiv 1 \pmod 8$ and $p^2 - \Delta q^2 = 4$. If q is odd, then $q^2 \equiv 1 \pmod 8$ implying $p^2 \equiv 5 \pmod 8$ which is impossible. Thus q , and hence p , are even, and the minimal positive solution is obtained from the fundamental solution of $p^2 - \Delta q^2 = 1$. Note that the fundamental solution of Pell's equation can be constructed by the method given in Section 1.III.

If $\Delta \equiv 5 \pmod 8$, the minimal positive solution may or may not be connected with the fundamental solution of $p^2 - \Delta q^2 = 1$. For example, $3 + \sqrt{5}$ is the minimal positive solution of $p^2 - 5q^2 = 4$, while $146 + 24\sqrt{37}$ is the minimal positive solution of $p^2 - 37q^2 = 4$. Proposition 1.12 can be used to decide which is the case.

Tables of minimal positive solutions to $x^2 - \Delta y^2 = 1$ and $x^2 - \Delta y^2 = 4$

appear on page 83 of Dickson (1939). References are given there to other tables.

Let $p_n + \sqrt{\Delta}q_n = 2\left[\left(p_1+\sqrt{\Delta}q_1\right)/2\right]^n$. Then

$$p_n + \sqrt{\Delta}q_n = \left(p_{n-1}+\sqrt{\Delta}q_{n-1}\right)\left(p_1+\sqrt{\Delta}q_1\right)/2 \tag{6}$$
$$= p_1\left(p_{n-1}+\sqrt{\Delta}q_{n-1}\right) - \left(p_1-\sqrt{\Delta}q_1\right)\left(p_{n-1}+\sqrt{\Delta}q_{n-1}\right)/2 .$$

Using (6),

$$\left(p_1-\sqrt{\Delta}q_1\right)\left(p_{n-1}+\sqrt{\Delta}q_{n-1}\right) = \left(p_{n-2}+\sqrt{\Delta}q_{n-2}\right)\left(p_1-\sqrt{\Delta}q_1\right)\left(p_1+\sqrt{\Delta}q_1\right)/2$$
$$= \left(p_{n-2}+\sqrt{\Delta}q_{n-2}\right)\left(p_1^2 - q_1^2\right)/2 .$$
$$= 2\left(p_{n-2}+\sqrt{\Delta}q_{n-2}\right) .$$

Thus

$$p_n + \sqrt{\Delta}q_n = p_1\left(p_{n-1}+\sqrt{\Delta}q_{n-1}\right) - \left(p_{n-2}+\sqrt{\Delta}q_{n-2}\right) . \tag{7}$$

Note that $p_0 + \sqrt{\Delta}q_0 = 2$.

9. Calculate $SU = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $U^{-1} = \begin{bmatrix} t & u \\ v & w \end{bmatrix}$.

10. Consider the sequence of automorphs of fS ,

$$A_n = \begin{bmatrix} \left(p_n-\delta' q_n\right)/2 & -\varepsilon' q_n \\ \gamma' q_n & \left(p_n+\delta' q_n\right)/2 \end{bmatrix} \quad \text{for} \quad n \geq 0 ,$$

where $p_n + \sqrt{\Delta}q_n$ is the solution of $p^2 - \Delta q^2 = 4$ determined by the recurrence relation (7).

After some matrix multiplication, SUA_nT^{-1} is in $GL_\lambda(2,\mathbb{Z})$ iff

$$\lambda \mid \left[(ct+dv)p_n/2 + (2\gamma' dt - \delta' ct - 2\varepsilon' cv + \delta' dv)q_n/2\right] .$$

Let

$$\Lambda_n = (ct+dv)p_n/2 + (2\gamma' dt - \delta' ct - 2\varepsilon' cv + \delta' dv)q_n/2 .$$

11. Compute Λ_0, Λ_1, Λ_2 , and so on. For $i \geq 2$, Λ_i can be computed

using the recurrence relation $\Lambda_i = p_1\Lambda_{i-1} - \Lambda_{i-2}$.

12. If $\lambda|\Lambda_i$, then $fSUA_iT^{-1} = \pm g$ and $f \sim_\lambda g$. //

If the values of Λ_j and Λ_{j+1} mod λ are the same as the values of Λ_i and Λ_{i+1} mod λ , where $i < j$, then the residues mod λ will recur.

13. If a recurrence of residues mod λ is observed without λ dividing any of the Λ_i's , then $f \not\sim_\lambda g$. //

At most $(\lambda-1)^2$ values of Λ_i will need to be computed.

Now

$$p_{-n} - \sqrt{\Delta}q_{-n} = p_1\left(p_{-(n-1)} - \sqrt{\Delta}q_{-(n-1)}\right) - \left(p_{-(n-2)} - \sqrt{\Delta}q_{-(n-2)}\right) .$$

This is shown in the analogous way by expanding

$$p_{-n} + \sqrt{\Delta}q_{-n} = 2\left[\left(p_1 - \sqrt{\Delta}q_1\right)/2\right]^n .$$

14-17. Steps 10-13 are repeated with the automorphs

$$A_{-n} = \begin{bmatrix} \left(p_{-n} - \delta' q_{-n}\right)/2 & -\varepsilon' q_{-n} \\ \gamma' q_{-n} & \left(p_{-n} + \delta' q_{-n}\right)/2 \end{bmatrix} , \quad n \geq 0 . \qquad //$$

$$\overline{A}_n = \begin{bmatrix} \left(-p_n + \delta' q_n\right)/2 & \varepsilon' q_n \\ -\gamma' q_n & \left(-p_n - \delta' q_n\right)/2 \end{bmatrix}$$

need not be considered, since $-\overline{A}_n$ is one of the A_n's or A_{-n}'s .

Next, the antiautomorphs connected with the solutions of $p^2 - \Delta q^2 = -4$ are considered.

By Proposition 1.14, if solutions of this equation exist, they can all be written in the form $p + \sqrt{\Delta}q = \pm 2\left[\left(p_1 + \sqrt{\Delta}q_1\right)/2\right]^{2n+1}$, where $p_1 + q_1\sqrt{\Delta}$ is the minimal positive solution, and n is an arbitrary integer.

18. Determine whether the equation $p^2 - \Delta q^2 = -4$ is soluble. If not, consider the next case of automorphs connected with the equation $x^2 - \Delta y^2 = 4$.

If so, compute the minimal positive solution $p_1 + \sqrt{\Delta} q_1$.

This equation is related to the Pellian equation $p^2 - \Delta q^2 = -1$, in the same way that $p^2 - \Delta q^2 = 4$ is related to $p^2 - \Delta q^2 = 1$. This is described after step 8. The method of continued fractions determines the minimal positive solution, and whether it exists, of $p^2 - \Delta q^2 = -1$.

Let $\bar{p} = \sqrt{\Delta}\,\bar{q} = (p_1 + \sqrt{\Delta} q_1)^2/4$.

Then $\bar{p} + \sqrt{\Delta}\,\bar{q}$ is a solution of Pell's equation.

Then all the solutions of $p^2 - \Delta q^2 = -4$ can be calculated by the recurrence relations

$$\pm(p_n + \sqrt{\Delta} q_n) = \pm 2\bar{p}(p_{n-1} + \sqrt{\Delta} q_{n-1}) \mp (p_{n-2} + \sqrt{\Delta} q_{n-2})$$

and

$$\pm(p_{-n} - \sqrt{\Delta} q_{-n}) = \pm 2\bar{p}(p_{-(n-1)} - \sqrt{\Delta} q_{-(n-1)}) \mp (p_{-(n-2)} - \sqrt{\Delta} q_{-(n-2)}) .$$

19-26. Steps 10-17 are repeated with the appropriate modifications. //

27. Consider the equations $x^2 - \Delta y^2 = \pm 4\gamma'\varepsilon'$.

Determine the number of classes of solutions, k say.

Proposition 1.12 is used to make this a finite test.

28. If $k = 0$, then $f \not\sim_\lambda g$. //

The remaining sequence of steps is done for each class of solutions, if no λ-equivalence is found.

29. Determine a representative $r + \sqrt{\Delta} s$ for the class.

Using Proposition 1.11, all solutions in that class can be written in the form

$$\pm(r+\sqrt{\Delta}s)(x_0+\sqrt{\Delta}y_0)^n ,$$

where $x_0 + \sqrt{\Delta}y_0$ is the fundamental solution of $x^2 - \Delta y^2 = 1$.

Let $r_n + \sqrt{\Delta}s_n = (r+\sqrt{\Delta}s)(x_0+\sqrt{\Delta}y_0)^n$. Then

$$\begin{aligned} r_n + \sqrt{\Delta}s_n &= (r_{n-1}+\sqrt{\Delta}s_{n-1})(x_0+\sqrt{\Delta}y_0) && (8)\\ &= 2x_0(r_{n-1}+\sqrt{\Delta}x_{n-1}) - (r_{n-1}+\sqrt{\Delta}s_{n-1})(x_0-\sqrt{\Delta}y_0) \\ &= 2x_0(r_{n-1}+\sqrt{\Delta}x_{n-1}) - (r_{n-2}+\sqrt{\Delta}s_{n-2})(x_0+\sqrt{\Delta}y_0)(x_0-\sqrt{\Delta}y_0) \text{ by (8)} \\ &= 2x_0(r_{n-1}+\sqrt{\Delta}s_{n-1}) - (r_{n-2}+\sqrt{\Delta}s_{n-2}) . && (9) \end{aligned}$$

Similarly,

$$r_{-n} - \sqrt{\Delta}s_{-n} = 2x_0(r_{-(n-1)}-\sqrt{\Delta}s_{-(n-1)}) - (r_{-(n-2)}-\sqrt{\Delta}s_{-(n-2)}) . \qquad (10)$$

Let

$$B_n = \begin{bmatrix} s_n & (r_n+\delta' s_n)/2\gamma' \\ (r_n-\delta' s_n)/2\varepsilon' & -s_n \end{bmatrix} .$$

30. Recall that $SU = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $T^{-1} = \begin{bmatrix} t & u \\ v & w \end{bmatrix}$ from step 9.

Then SUB_nT^{-1} is in $GL_\lambda(2,\mathbb{Z})$ iff

$$\lambda \mid \{(cv/2\gamma')+(td/2\varepsilon')\}r_n + \{ct-dv+(\delta' cv/2\gamma')-(\delta' dt/2\varepsilon')\}s_n .$$

Let

$$K_n = \gamma'\varepsilon'\left[\{(cv/2\gamma')+(dt/2\varepsilon')\}r_n+\{ct-dv+(\delta' cv/2\gamma')-(\delta' dt/2\varepsilon')\}s_n\right] .$$

Then $f \sim_\lambda g$ if $2\gamma' \mid r_n+\delta' s_n$, $2\varepsilon' \mid r_n-\delta' s_n$, and $\lambda\gamma'\varepsilon' \mid K_n$.

Because of the recurrence relations (9) and (10), whether this occurs can be determined by computing a finite number of values of $r_n + \delta' s_n$, $r_n - \delta' s_n$ and K_n .

31. Perform the requisite divisibility checks to determine whether $f \sim_\lambda g$. // □

VI. Examples

Some computations are presented to demonstrate how the methods of this chapter could be applied to answering questions about groups in $T(4, 2)$.

Consider the two groups presented in Proposition B of Grunewald and Scharlau (1979).

$$G_1 = \langle g_1, g_2, g_3, g_4; [g_1, g_2] = [g_3, g_4] = \emptyset, [g_2, g_3] = [g_1, g_4],$$
$$[g_2, g_4] = [g_1, g_3]^{-5}, \text{ class } 2\rangle ,$$

and

$$G_2 = \langle g_1, g_2, g_3, g_4; [g_1, g_2] = [g_3, g_4] = \emptyset,$$
$$[g_2, g_3] = [g_1, g_4]^2[g_1, g_3]^{-1}, [g_2, g_4] = [g_1, g_4][g_1, g_3]^{-3}, \text{ class } 2\rangle .$$

The presentations as written are not restricted canonical presentations, nor even canonical presentations. However canonical presentations can easily be given for these groups. Relabel the generators as follows:-

$$g_1 \to a_1 , \quad g_2 \to a_4 , \quad g_3 \to a_2 , \quad g_4 \to a_3 .$$

Introduce b_1, b_2 as a basis for $I(G_i')/G_i'$, $i = 1, 2$.

$$\overline{G}_1 = \langle a_1, a_2, a_3, a_4, b_1, b_2; [a_2, a_1] = b_1, [a_3, a_1] = b_2,$$
$$[a_4, a_1] = [a_3, a_2] = \emptyset, [a_4, a_2] = b_2^{-1}, [a_4, a_3] = b_1^5, \text{ canonical}\rangle ,$$

$$\tilde{G}_2 = \langle a_1, a_2, a_3, a_4, b_1, b_2; [a_2, a_1] = b_1, [a_3, a_1] = b_2,$$
$$[a_4, a_1] = [a_3, a_2] = \emptyset, [a_4, a_2] = b_1b_2^{-2}, [a_4, a_3] = b_1^3b_2^{-1}, \text{ canonical}\rangle .$$

$\overline{G}_1$ is a restricted canonical presentation, and an easy manipulation gives

$$\overline{G}_2 = \langle a_1, a_2, a_3, a_4, b_1, b_2; [a_2, a_2] = b_1, [a_3, a_1] = b_2,$$
$$[a_4, a_1] = [a_3, a_2] = \emptyset, [a_4, a_2] = b_2^{-2}, [a_4, a_3] = b_1^3b_2^{-2}, \text{ canonical}\rangle$$

as a restricted canonical presentation equivalent to G_2 .

Both $I(G_1')/G_1'$ and $I(G_2')/G_2'$ are trivial. So

$$\alpha = \beta = \lambda = 1 ,$$

$$\mathrm{Pf}(\overline{G}_1) = 5x^2 + y^2 ,$$

$$\mathrm{Pf}(\overline{G}_2) = 3x^2 - 2xy + 2y^2 ,$$

$$\Delta(\mathrm{Pf}(\overline{G}_1)) = -4.5 = -20 ,$$

$$\Delta(\mathrm{Pf}(\overline{G}_2)) = (-2)^2 - 4.3.2 = -20 ,$$

$$o(\mathrm{Pf}(\overline{G}_1)) = 1 = o(\mathrm{Pf}(\overline{G}_2)) .$$

Grunewald and Scharlau prove that the groups have the same finite quotients. So they satisfy Conjecture 4.6.

The Pfaffians are definite forms.

The reduced form equivalent to $5x^2 + y^2$ is $x^2 + 5y^2$. The reduced form equivalent to $3x^2 - 2xy + 2y^2$ is $2x^2 + 2xy + 3y^2$. These are not identical, so the groups are non-isomorphic.

Recall the family of groups given at the beginning of Section II,

$$P_k = \langle a_1, a_2, a_3, a_4, b_1, b_2; [a_2, a_1] = b_1, [a_3, a_1] = b_2,$$

$$[a_4, a_1] = [a_3, a_2] = \emptyset, [a_4, a_2] = b_2^{-k^2},$$

$$[a_4, a_3] = b_1 b_2^{2k}, \text{ restricted canonical} \rangle .$$

They are isomorphic, on inspection, iff their Pfaffians are equivalent.

$\mathrm{Pf}(P_k) = x^2 + 2kxy + k^2y^2$, which has discriminant 0 for all k.

But $(1, 2k, k^2) \sim (1, 0, 0)$ by applying Lemma 4.7 *(iii)* repeatedly. So the groups are all isomorphic.

The index of $\mathrm{GL}_\lambda(2, \mathbb{Z})$ in $\mathrm{GL}(2, \mathbb{Z})$ is useful if one wishes to construct a list of isomorphism types of groups in $T(4, 2)$. The following result is given in Chapter VII of Morris Newman (1972).

PROPOSITION 4.22. $|GL(2,\mathbb{Z}) : \Gamma_1(\lambda)| = \lambda \prod_{p|\lambda} (1+(1/p))$.

It is helpful, also, when constructing such a list to have a complete set of coset representatives for $GL_\lambda(2,\mathbb{Z})$ in $GL(2,\mathbb{Z})$. Coset representatives can be easily given when λ is a prime.

PROPOSITION 4.23. *The matrices* $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$, $0 \le k < p$, *form a complete set of right coset representatives for* $GL_p(2,\mathbb{Z})$ *in* $GL(2,\mathbb{Z})$ *when* p *is a prime.*

Proof. Let $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ be in $GL(2,\mathbb{Z})$. If $p|d$, then

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} b & a \\ d & c \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} b & a \\ d & c \end{bmatrix}$ is in $GL_p(2,\mathbb{Z})$. Otherwise $(p, d) = 1$. Choose α and β such that $\alpha p + \beta d = c$, where $0 \le \beta < p$. Then

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a-\beta b & b \\ \alpha p & d \end{bmatrix}\begin{bmatrix} 1 & 0 \\ \beta & 1 \end{bmatrix} .$$

Thus a complete set of right coset representatives for $GL_p(2,\mathbb{Z})$ in $GL(2,\mathbb{Z})$ can be chosen from the $p + 1$ elements $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$, $0 \le k < p$.

It remains to show that these elements lie in distinct cosets.

Suppose

$$\begin{bmatrix} t & u \\ vp & w \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix} .$$

Then $t = 0$, which is impossible since $tw - puv = \pm 1$. So $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is in a coset distinct from the others.

Suppose $\begin{bmatrix} t & u \\ vp & w \end{bmatrix}\begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ k' & 1 \end{bmatrix}$, where $0 \le k, k' < p$. Then

$$t + uk = 1 \ , \quad u = 0 \ ,$$

$$vp + wk = k' \ , \quad w = 1 \ .$$

Thus $vp + k = k'$. By assumption then, $v = 0$, and $k = k'$. □

This proposition gives a proof of Proposition 4.22 when λ is a prime.

Proposition 4.23 is now used to list all isomorphism types of groups with $I(G')/G' \cong C_2$ and with Pfaffian of discriminant 5.

From the first condition $\alpha = 1$, $\beta = 2$, and hence $\lambda = 2$. Let the Pfaffian be $(\gamma, \delta, \varepsilon)$. Then by assumption, $\delta^2 - 4\gamma\varepsilon = 5$. Since the form is reduced, $5 \ge \delta^2 + 4\delta^2 = 5\delta^2$.

So $\delta^2 \le 1$ and $\delta = 0$ or 1. But $\delta = 0$ is not possible. Thus the only reduced form is $(1, 1, -1)$.

By Proposition 4.22, $|GL(2,\mathbb{Z}) : GL_2(2,\mathbb{Z})| = 3$. So the orbit of the form under $GL_2(2,\mathbb{Z})$ splits into at most 3 orbits, and there are at most 3 groups with the desired properties. By Proposition 4.23, $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ are a complete set of coset representatives for $GL_2(2,\mathbb{Z})$ in $GL(2,\mathbb{Z})$.

Now $(1, 1, -1)\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \sim_2 (1, 1, -1)$ iff $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{-1} A$ is in $GL_2(2,\mathbb{Z})$ for some element A in $\mathrm{Autom}\{(1, 1, -1)\}$ by Lemma 4.20.

By Proposition 4.19, one family of elements of $\mathrm{Autom}\{(1, 1, -1)\}$ is obtained from solutions of $x^2 - 5y^2 = 4$. The minimal positive solution of this equation is $3 + \sqrt{5}$.

The automorph corresponding to this solution is $A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$. But

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{-1} A^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} \in GL_2(2, \mathbb{Z}) .$$

Thus $(1, 1, -1)\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \sim_2 (1, 1, -1)$.

Similarly $(1, 1, -1)\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \sim_2 (1, 1, -1)$ iff $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} B$ is in $GL_2(2, \mathbb{Z})$ for some element B in $\mathrm{Autom}\big((1, 1, -1)\big)$. But

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} A = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \in GL_2(2, \mathbb{Z}) .$$

Thus there is only one isomorphism class of groups satisfying the condition that $I(G')/G' \cong C_2$ and the Pfaffian of a restricted canonical presentation for G has discriminant 5. A presentation for this group is

$$\Big\langle a_1, a_2, a_3, a_4, b_1, b_2; [a_2, a_1] = b_1, [a_3, a_1] = b_2^2,$$
$$[a_4, a_1] = [a_3, a_2] = \emptyset, [a_4, a_2] = b_2^2,$$
$$[a_4, a_3] = b_1 b_2^2, \text{ restricted canonical} \Big\rangle .$$

CHAPTER 5

ISOLATED GROUPS

The theory developed for torsionfree groups in Chapters 3 and 4 can be adapted for another family of finitely presented nilpotent groups of class 2 . The first section introduces these groups, and also relational presentations. These presentations have the same role here that canonical presentations had in the discussion of torsionfree groups. The second section gives a bijection between canonical and relational presentations, while the final section restates the classification results of the earlier chapters in the current context.

All groups considered are nilpotent of class 2 .

I. Relational Presentations

The exposition of this section closely parallels the exposition of canonical presentations in Section 3.I.

A group G is *isolated* if G/G' is torsionfree. An equivalent definition is that $I_G(G') = G'$, that is the commutator subgroup is isolated.

Let F_d denote the free nilpotent group of class 2 on d generators.

LEMMA 5.1. *Suppose G is a d-generator group isomorphic to F_d/N . Then G is isolated iff $N \leq F_d'$.*

Proof. $G/G' \cong (F_d/N)/(F_d/N)' = (F_d/N)/(F_d'N/N) \cong F_d/F_d'N$. If $N \leq F_d'$, then $G/G' \cong F_d/F_d'$, which is torsionfree. If G is isolated, then G/G' is a free abelian group of rank d . So $G/G' \cong F_d/F_d'$ and $N \leq F_d'$. □

For the rest of this section, G is a d-generator group isomorphic to F_d/N.

Now N is a subgroup of F_d', which is a free abelian group of rank $\binom{d}{2}$. Thus N is a free abelian group of rank s say, where $s \leq \binom{d}{2}$. Then $h(G) = d + \binom{d}{2} - s$.

Let $I(d, s)$ denote the family of d-generator isolated groups with Hirsch number $d + \binom{d}{2} - s$. The only group in $I(d, 0)$ is F_d.

DEFINITION 5.2. Choose elements $a_1, \ldots, a_d$ of F_d that generate F_d. Choose elements $v_1, \ldots, v_s$ of F_d' such that $v_1, \ldots, v_s$ form a basis of N. Then G can be presented by

$$\Big\langle a_1, \ldots, a_d;\ v_k = \prod_{1 \leq i < j \leq d} [a_j, a_i]^{\alpha(i,j,k)} = \emptyset,\ 1 \leq k \leq s,$$

$$\text{class } 2,\ \alpha(i, j, k) \in \mathbb{Z} \Big\rangle .$$

Such a presentation will be called a *relational* presentation.

The v_k's must be independent in F_d' for the presentation to be relational. The condition 'relational' in a presentation will imply the class 2 condition, that $\alpha(i, j, k) \in \mathbb{Z}$, and that the v_k's are independent.

Relational presentations clearly present isolated groups. As for canonical presentations, there are in general many relational presentations for a given group. The rest of this section establishes conditions for when two relational presentations present isomorphic groups.

The convention is again adopted that $\alpha(i, i, k) = 0$ and $\alpha(i, j, k) = -\alpha(j, i, k)$ for $i > j$.

A $d \times d$ skew-symmetric matrix can be associated with a given

relational presentation V. The entries of the matrix, which is denoted M_V, are linear homogeneous polynomials with integer coefficients, and are defined by

$$M_V(i, j) = \sum_{k=1}^{s} \alpha(i, j, k)x_k .$$

The notation for these matrices is that used in Chapter 3.

The parameters $\alpha(i, j, k)$ clearly depend on the choice of a_i's and v_k's in Definition 5.2. Let $b_1, \ldots, b_d$ be another set of generators of F_d. Clearly $b_1F'_d, \ldots, b_dF'_d$ form a basis for F_d/F'_d.

By Theorem 1.2, there is an element T of $GL(d, \mathbb{Z})$ such that $a_iF'_d = \left\{\prod_{m=1}^{d} b_m^{T(i,m)}\right\}F'_d$. Then

$$[a_j, a_i] = \left[\prod_{n=1}^{d} b_n^{T(j,n)}x, \prod_{m=1}^{d} b_m^{T(i,m)}y\right]$$

for appropriate elements x, y in F'_d

$$= \prod_{m=1}^{d}\prod_{n=1}^{d} [b_n, b_m]^{T(i,m)T(j,n)} .$$

Let $w_1, \ldots, w_s$ form another basis for N. By Theorem 1.2, there is an element S of $GL(s, \mathbb{Z})$ such that $w_k = \prod_{L=1}^{s} v_L^{S(k,L)}$.

Let $\overline{V}$ be the relational presentation for G given in terms of the b_i's and w_j's, and let $M_{\overline{V}}(i, j) = \sum_{k=1}^{s} \beta(i, j, k)x_k$.

Now from the above,

$$w_k = \prod_{L=1}^{s} \prod_{1\le i<j\le d} \prod_{m=1}^{d}\prod_{n=1}^{d} [b_n, b_m]^{T(i,m)T(j,n)\alpha(i,j,L)S(k,L)} .$$

Thus $\beta(i, j, k)$ is the exponent of $[b_j, b_i]$ in the above expression.

Relabelling summation variables appropriately,

$$\beta(i, j, k) = \sum_{L=1}^{s} \sum_{1\le m<n\le d} \alpha(m, n, L)T(i, m)T(j, n)S(k, L)$$

$$- \sum_{L=1}^{s} \sum_{1\le m<n\le d} \alpha(m, n, L)T(i, n)T(j, m)S(k, L)$$

$$= \sum_{L=1}^{s} \sum_{1\le m<n\le d} \alpha(m, n, L)T(i, m)T(j, n)S(k, L)$$

$$+ \sum_{L=1}^{s} \sum_{1\le m<n\le d} \alpha(n, m, L)T(i, n)T(j, m)S(k, L) .$$

Interchanging m and n in the second sum and combining gives

$$\sum_{L=1}^{s} \sum_{m=1}^{d} \sum_{n=1}^{d} \alpha(m, n, L)T(i, m)T(j, n)S(k, L) .$$

In section 3.I, it was shown that

$$TM_V^S T^t(i, j) = \sum_{k=1}^{s} \sum_{L=1}^{s} \sum_{m=1}^{d} \sum_{n=1}^{d} \alpha(m, n, L)T(i, m)T(j, n)S(k, L)x_L .$$

Thus $TM_V^S T^t = M_{\overline{V}}$.

THEOREM 5.3. *Let V, W be two relational presentations of groups in $I(d, s)$. Then V, W present isomorphic groups iff there exist elements S of $\mathrm{GL}(s, \mathbb{Z})$ and T of $\mathrm{GL}(d, \mathbb{Z})$ such that $M_W = TM_V^S T^t$.*

Proof. Let

$$V = \left\langle a_1, \ldots, a_d; v_k = \prod_{1\le i<j\le d} [a_j, a_i]^{\alpha(i,j,k)} = \emptyset,\ 1 \le k \le s, \text{ relational} \right\rangle$$

and

$$W = \left\langle b_1, \ldots, b_d; w_k = \prod_{1\le i<j\le d} [b_j, b_i]^{\beta(i,j,k)} = \emptyset,\ 1 \le k \le s, \text{ relational} \right\rangle$$

present G, H respectively, where $G \cong F_d/N$ and $H \cong F_d/L$.

IF Let $\overline{a_i} = \prod_{m=1}^{d} a_i^{T^{-1}(i,m)}$ for $1 \le i \le d$, and $\overline{v_k} = \prod_{L=1}^{s} v_L^{S(k,L)}$

for $1 \le k \le s$. Clearly $\bar{a}_1, \ldots, \bar{a}_d$ generate G and by Theorem 1.2, $\bar{v}_1, \ldots, \bar{v}_s$ form a basis for N . Thus a relational presentation $\bar{V}$ can be written for G in terms of the $\bar{a}_i$'s and $\bar{v}_j$'s .

By the above discussion, $M_{\bar{V}} = TM_V^S T^t$ which equals M_W by assumption. There is an obvious isomorphism mapping G to H .

ONLY IF Suppose $\theta : F_d/N \to F_d/L$ is an isomorphism. We construct an automorphism of F_d mapping N to L . Define a map ψ on $a_1, \ldots, a_d$ such that

$$(a_i\psi)L = (a_iN)\theta \ .$$

ψ can be extended to an endomorphism of F_d by the relative freeness of F_d .

Since θ is an isomorphism, there exist c_i's , $1 \le i \le d$, such that

$$(c_iN)\theta = a_iL \ .$$

Then $(c_i\psi)L = (c_iN)\theta = a_iL$. So the $c_i\psi$'s generate F_d together with L . But $L \le F_d'$ by Lemma 5.1 and is thus omissible. So $F_d\psi = F_d$ and ψ is onto. Since F_d is hopfian, it follows that ψ is an automorphism of F_d .

Let $v \in N$. Then $(v\psi)L = (vN)\theta = N\theta = L$, showing that $N\psi \le L$.

Now θ^{-1} is an isomorphism from F/L to F/N . Similarly, an automorphism η of F_d can be found such that $L\eta \le N$. Thus $L\eta\psi \le N\psi \le L$.

Now

$$F/L \cong (F/L)\eta \cong F/L\eta \cong (F/L\eta)\psi \cong F/L\eta\psi \ .$$

If $L\eta\psi < L$, then F/L is isomorphic to a proper quotient of itself, which

is impossible, since finitely generated nilpotent groups are hopfian. Thus $N\psi = L$. Let $\overline{W}$ be the relational presentation of H relative to $a_1\psi, \ldots, a_d\psi$ and $v_1\psi, \ldots, v_s\psi$. Then

$$v_k\psi = \left(\prod_{1 \le i < j \le d} [a_j, a_i]^{\alpha(i,j,k)} \right)\psi$$

$$= \prod_{1 \le i < j \le d} [a_j\psi, a_i\psi]^{\alpha(i,j,k)}$$

and so $M_{\overline{W}} = M_V$.

By the earlier discussion, elements S of $\mathrm{GL}(s, \mathbb{Z})$ and T of $\mathrm{GL}(d, \mathbb{Z})$ can be found such that $M_{\overline{W}} = T^{-1} M_W^{S^{-1}} (T^{-1})^t$.

Thus $M_W = T M_{\overline{W}}^{S} T^t = T M_V^{S} T^t$. □

II. An Association Between Canonical and Relational Presentations

The close parallel between Section 3.I concerned with canonical presentations and the previous section on relational presentations is made more precise in this section. A canonical presentation can be associated with each relational presentation, and *vice versa*. This association can be formalised in several ways to give a duality between groups in $T(d, s)$ and groups in $I(d, s)$. I thank Dr L.G. Kovács and Dr C. Leedham-Green for showing me two methods. The duality appears analogous to that discussed by Gauger (1973) and (1974) and Scheuneman (1967) in the context of 2-step nilpotent Lie algebras.

Let P be the canonical presentation

$$\left\langle a_1, \ldots, a_d, b_1, \ldots, b_s; [a_j, a_i] = \prod_{k=1}^{s} b_k^{\alpha(i,j,k)}, \ 1 \le i < j \le d, \right.$$

$$\left. \text{canonical} \right\rangle .$$

A presentation for an isolated group, to be denoted $D(P)$, can be associated with P - namely

$$\Big\langle a_1, \ldots, a_d;\ v_k = \prod_{1\le i<j\le d} [a_j, a_i]^{\alpha(i,j,k)} = \emptyset,$$

$$1 \le k \le s,\ \alpha(i, j, k) \in \mathbb{Z},\ \text{class } 2\Big\rangle .$$

Because P is a canonical presentation, for G say, the b_k's form a basis for $I(G')$. Thus the v_k's are independent elements of F_d', and $D(P)$ is a relational presentation.

For example, if

$$P = \langle a_1, a_2, b;\ [a_2, a_1] = b^{\alpha},\ \text{canonical}\rangle ,$$

then

$$D(P) = \langle a_1, a_2;\ [a_2, a_1]^{\alpha} = \emptyset,\ \text{relational}\rangle .$$

Clearly $M_P = M_{D(P)}$. Also, if P presents a group with Hirsch number $d + s$, then $D(P)$ presents a group with Hirsch number $d + \binom{d}{2} - s$.

Conversely, let V be the relational presentation

$$\Big\langle a_1, \ldots, a_d;\ v_k = \prod_{1\le i<j\le d} [a_j, a_i]^{\beta(i,j,k)} = \emptyset,\ 1 \le k \le s,\ \text{relational}\Big\rangle .$$

Then the following presentation, denoted $\overline{D}(V)$, can be associated with V:

$$\Big\langle a_1, \ldots, a_d, b_1, \ldots, b_s;\ [a_j, a_i] = \prod_{k=1}^{s} b_k^{\beta(i,j,k)},$$

$$\beta(i, j, k) \in \mathbb{Z},\ \text{class } 2\Big\rangle .$$

Because the b_k's are independent, $\overline{D}(V)$ is a canonical presentation. Again, $M_V = M_{\overline{D}(V)}$.

Now $M_P = M_{D(P)} = M_{\overline{D}(D(P))}$, and thus P and $\overline{D}(D(P))$ present

isomorphic groups. Similarly, V and $D(\bar{D}(V))$ present isomorphic groups.

THEOREM 5.4. *Let P, Q be two canonical presentations. Then P, Q present isomorphic groups iff $D(P)$, $D(Q)$ present isomorphic groups.*

Proof. ONLY IF Let P and Q present isomorphic groups. By Theorem 3.3, there are integer matrices S and T such that $M_Q = TM_P^S T^t$. By the discussion above, $M_Q = M_{D(Q)}$ and $M_P = M_{D(P)}$. Thus $M_{D(Q)} = TM_{D(P)}^S T^t$, and by Theorem 5.3, $D(P)$ and $D(Q)$ present isomorphic groups.

IF Let $D(P)$ and $D(Q)$ present isomorphic groups. By Theorem 5.3, there are integer matrices S and T such that $M_{D(Q)} = TM_{D(P)}^S T^t$. But $M_{D(Q)} = M_Q$, and $M_{D(P)} = M_P$, and so $M_Q = TM_P^S T^t$. By Theorem 3.3, P and Q present isomorphic groups. □

III. 'Dual' Classification Results

Because of Theorem 5.4, the results of Section 3.II can be restated for isolated groups.

PROPOSITION 5.5. *Every group in $I(d, 1)$ has a presentation of the form*

$$\langle a_1, \ldots, a_d; [a_2, a_1]^{h_1}[a_4, a_3]^{h_2} \ldots [a_{2r}, a_{2r-1}]^{h_r} = \emptyset,\ r = \lfloor d/2 \rfloor,$$
$$h_1 > 0,\ h_i \geq 0,\ 2 \leq i \leq r,\ h_{i-1} | h_i,\ 2 \leq i \leq r,\ \textit{relational} \rangle .$$

Different values of h_i give nonisomorphic groups.

This is essentially Proposition D of Grunewald and Scharlau (1979).

PROPOSITION 5.6. *Every group in $I(3, 2)$ has a presentation of the form*

$$\langle a_1, a_2, a_3; [a_2, a_1]^{\alpha} = \emptyset,\ [a_3, a_1]^{\beta} = \emptyset,\ \alpha, \beta > 0,\ \alpha | \beta,\ \textit{relational} \rangle .$$

Different values of α, β give nonisomorphic groups.

PROPOSITION 5.7. *Every group in* $I(3, 3)$ *has a presentation of the form*

$$\langle a_1, a_2, a_3; [a_2, a_1]^{\alpha} = \emptyset, [a_3, a_1]^{\beta} = \emptyset, [a_3, a_2]^{\gamma} = 0,$$

$$\alpha, \beta, \gamma > 0, \alpha|\beta|\gamma, \textit{relational}\rangle .$$

Different values of α, β, γ *give nonisomorphic groups.*

The 'dual' of Proposition 4.1 can also be stated for relational presentations.

PROPOSITION 5.8. *Every group in* $I(4, 2)$ *has a presentation of the form*

$$\langle a_1, a_2, a_3, a_4; [a_2, a_1]^{\alpha}[a_4, a_3]^{\gamma} = \emptyset, [a_3, a_1]^{\beta}[a_4, a_2]^{-\varepsilon}[a_4, a_3]^{\delta} = \emptyset,$$

$$\alpha, \beta, \gamma, \delta, \varepsilon \in \mathbb{Z}, \alpha, \beta > 0, \alpha|\beta,\gamma , \beta|\delta,\varepsilon , \textit{relational}\rangle .$$

Call such a presentation a *restricted relational presentation.* As in Section 4.II, a binary quadratic form can be associated with a restricted relational presentation. If V is the restricted relational presentation of Proposition 5.8, define $\mathrm{Pf}(V)$, called the *Pfaffian* of V, to be $\gamma' x^2 + \delta' xy + \varepsilon' y^2$, where $\gamma' = \gamma/\alpha$, $\delta' = \delta/\beta$, $\varepsilon' = \varepsilon/\beta$. Theorems 4.2 and 5.4 can be combined to prove that two restricted relational presentations present isomorphic groups iff their commutator subgroups are isomorphic and their Pfaffians are λ-equivalent. An explicit isomorphism between the two isolated groups is then given by combining Theorems 3.3, 4.2 and 5.3.

CHAPTER 6

COMPUTING PRESENTATIONS

This short chapter discusses how the special presentations of Chapters 3-5 may be obtained from arbitrary finite presentations. The basic step, which is well-known, is to construct first a special presentation from which the isomorphism type of the commutator quotient group is immediate. Most of the terminology of this chapter is based on that of Magnus, Karrass and Solitar (1976).

Let $W = \prod_{i=1}^{r} a_{\nu_i}^{\alpha(i)}$ be a word in $a_1, \ldots, a_n$, where $\alpha(i) \in \mathbb{Z}$, $1 \leq i \leq r$.

The *exponent sum* of W on a_ν, denoted $\sigma_\nu(W)$, is the integer

$$\sigma_\nu(W) = \sum_{\nu_i=\nu} \alpha(i) .$$

For example, if $W = a_2^2 a_1^{-1} a_2 a_1^3 a_2^{-2}$, then $\sigma_1(W) = 2$ and $\sigma_2(W) = 1$.

Recall that F_d is the free nilpotent group of class 2 on d generators.

LEMMA 6.1. *Let F_d be generated by $a_1, \ldots, a_d$ and let W be a word in $a_1, \ldots, a_d$. Then W is in F_d' iff $\sigma_i(W) = 0$, $1 \leq i \leq d$.*

Proof. This lemma is true when F_d is replaced by the free group on d generators - see Problem 2 of Section 2.2 of Magnus, Karrass and Solitar (1976). It is proved here only in the context of nilpotent groups of class 2.

ONLY IF If W is in F_d', then it is a product of commutators. Each commutator has zero exponent sum, so $\sigma_i(W) = 0$ for $1 \leq i \leq d$.

IF We prove more than is needed, since the form of W will be used later.

Because G' is central, W can be collected into the form

$$\prod_{k=1}^{d} a_k^{\sigma(k)} \prod_{1 \le i < j \le d} [a_j, a_i]^{\alpha(i,j)} ,$$

where $\sigma(k) = \sigma_k(W)$, and $\alpha(i, j) \in \mathbb{Z}$. By assumption, all the exponent sums are zero, and W is in F_d'. □

This can easily be done - for example using the algorithm of Section 3.I.

Let $P = \langle a_1, \ldots, a_n; W_i = \emptyset, 1 \le i \le n \rangle$ be a presentation for a group G. A *relation matrix* for P is the $m \times n$ integer matrix M defined by $M(i, j) = \sigma_j(W_i)$. The Smith normal form of this relation matrix determines the isomorphism type of G/G'. Note that it is straightforward to compute the relation matrix of a presentation. This is done in practice, for example, by an interface program between the Reidemeister-Schreier program of Havas (1974) and the program described in Chapter 2.

THEOREM 6.2. *Given a finite presentation for a group G, a set of generators $y_1, \ldots, y_n$ and a set of defining relations $V_1 = \emptyset, \ldots, V_{n+m} = \emptyset$ can be constructed having the form $V_i = y_i^{\rho(i)} Q_i$, $i = 1, \ldots, n$, where Q_i, $1 \le i \le n$, and V_{j+n}, $1 \le j \le m$, have zero exponent sum on each y_i. Further, $\rho(i) \ge 0$ and $\rho(i-1) \mid \rho(i)$ for $2 \le i \le n$.*

Such a presentation of G is called a *pre-abelian presentation*, following Magnus, Karrass and Solitar (1976). This theorem is essentially their Theorem 3.5 and a proof appears there. The steps of the proof are to apply elementary row and column operations to the relation matrix, converting it into its Smith normal form. Corresponding to each elementary

column operation, Tietze transformations can be performed on the relations. A different set of generators is chosen for the group, corresponding to elementary row operations. It is straightforward to adapt the basic algorithm of Chapter 2 to find the y_i's and V_j's. In Chapter Seven of Hartley and Hawkes (1970), this is explicitly done for abelian groups. The isomorphism type of G/G' is immediate from a pre-abelian presentation.

For a nilpotent group G of class 2, a pre-abelian presentation can be simplified. Using Lemma 6.1, the pre-abelian presentation can be chosen to have the form

$$\Big\langle y_1, \ldots, y_n;\ y_k^{\rho(k)} = \prod_{1\le i<j\le n} [y_j, y_i]^{\alpha(i,j,k)},\ 1 \le k \le n,$$

$$V_L = \prod_{1\le i<j\le n} [y_j, y_i]^{\beta(i,j,L)} = \emptyset,\ 1 \le L \le m,$$

$$\rho(k),\ \alpha(i, j, k),\ \beta(i, j, L) \in \mathbb{Z},\ \rho(k) \ge 0,\ 1 \le k \le n,$$

$$\rho(k-1) \mid \rho(k),\ 2 \le k \le n,\ \text{class } 2\Big\rangle . \quad (*)$$

Suppose $\rho(k) = 1$. Then y_k is in $G' \le Z(G)$. Thus the relation $y_k = \prod_{1\le i<j\le n} [y_j, y_i]^{\alpha(i,j,k)}$ can be used to eliminate y_k from the presentation, using a Tietze transformation of type T4.

Suppose G is isolated. Then G/G' is torsionfree, and so all the $\rho(k)$'s are 0 or 1. Thus for an isolated group the pre-abelian presentation can be chosen to have the form

$$\Big\langle y_1, \ldots, y_d;\ V_k = \prod_{1\le i<j\le d} [y_j, y_i]^{\alpha(i,j,k)} = \emptyset,\ 1 \le k \le m,$$

$$\alpha(i, j, k) \in \mathbb{Z},\ \text{class } 2\Big\rangle .$$

Let F_d be the free nilpotent group of class 2 on the d generators $y_1, \ldots, y_d$. Then the above presentation is relational if the m relations $V_k = \emptyset$ are replaced by s relations $v_k = \emptyset$, $1 \le k \le s$, where the v_k's

form a basis for the subgroup of F_d' generated by the V_k's .

Consider the $m \times \binom{d}{2}$ integer matrix M formed by setting the kth row of the matrix to be the $\binom{d}{2}$ exponents, $\alpha(i, j, k)$ for $1 \leq i < j \leq d$, of $[a_j, a_i]$ in V_k . Note that $\alpha(i, j, k)$ is in the same column for $1 \leq k \leq m$. Elementary row operations can be applied to the matrix to transform it into a triangular form where $M(k, L) = 0$ if $k > L$. The basic algorithm of Chapter 2 can be suitably adapted to do this. Reinterpreting this matrix gives the v_k's . Thus given a finite presentation for a group and the information that it is an isolated nilpotent group of class 2 , a relational presentation can be computed for the group.

The procedure for obtaining a canonical presentation from a pre-abelian presentation is only a little more involved. It is based on an algorithm given on pages 303-304 of Knuth (1969) to solve a system of linear Diophantine equations.

Let G be a torsionfree nilpotent group of class 2 , presented by the presentation (*). Suppose $\rho(k) > 1$. Then $y_k^{\rho(k)}$ is in $G' \leq Z(G)$, and $[y_k, g]^{\rho(k)} = \left[y_k^{\rho(k)}, g\right] = \emptyset$, for any element g of G . Because G is torsionfree, $[y_k, g] = \emptyset$. Thus G has a pre-abelian presentation of the form

$$\Big\langle a_1, \ldots, a_d, b_1, \ldots, b_t;\ b_k^{\rho(k)} = \prod_{1 \leq i < j \leq d} [a_j, a_i]^{\alpha(i,j,k)},\ 1 \leq k \leq t,$$

$$W_L = \prod_{1 \leq i < j \leq d} [a_j, a_i]^{\beta(i,j,L)} = \emptyset,\ 1 \leq L \leq m,$$

$$\rho(k),\ \alpha(i, j, k),\ \beta(i, j, L) \in \mathbb{Z},\ \rho(k) \geq 2,\ \text{class } 2\Big\rangle .$$

For ease of description of the algorithm, $I(G')$ will be written additively.

1. Introduce $\binom{d}{2}$ elements $c_{ij} = [a_j, a_i]$, $1 \le i < j \le d$. Then b_k for $1 \le k \le t$, and c_{ij} for $1 \le i < j \le d$, generate $I(G')$. There are $m + t$ relations between these $\binom{d}{2} + t$ generators.

2. Find a non-zero coefficient of smallest absolute value occurring in the relations. Suppose it occurs in the relation

$$\sum_{k=1}^{t} \alpha_k b_k + \sum_{1 \le i < j \le d} \beta(i, j) c_{ij} = 0 . \qquad (1)$$

For convenience, assume α_1 is the minimal coefficient. The procedure is similar for the other coefficients.

3. If $\alpha_1 < 0$, replace relation (1) by

$$\sum_{k=1}^{t} \bar{\alpha}_k b_k + \sum_{1 \le i < j \le d} \bar{\beta}(i, j) c_{ij} = 0 ,$$

where $\bar{\alpha}_k = -\alpha_k$, $1 \le k \le t$, and $\bar{\beta}(i, j) = -\beta(i, j)$, $1 \le i < j \le d$.

As for other algorithm descriptions, α_k and $\beta(i, j)$ will refer to current values of the coefficients, rather than their initial values in the relations.

4. If $\alpha_1 > 1$, introduce a new generator

$$\bar{b}_1 = \sum_{k=1}^{t} \bar{\alpha}_k b_k + \sum_{1 \le i < j \le d} \bar{\beta}(i, j) c_{ij} , \qquad (2)$$

where $\bar{\alpha}_k = \lfloor \alpha_k / \alpha_1 \rfloor$, $1 \le k \le t$, and $\bar{\beta}(i, j) = \lfloor \beta(i, j) / \alpha_1 \rfloor$, $1 \le i < j \le d$. Note that $\bar{\alpha}_1 = 1$.

5. Use (2) to eliminate b_1 in favour of $\bar{b}_1$ from the system of relations. Abusing notation, write b_1 for $\bar{b}_1$.

6. Replace relation (1) by

$$\sum_{k=1}^{t} \tilde{\alpha}_k b_k + \sum_{1 \le i < j \le d} \tilde{\beta}(i, j) c_{ij} = 0 ,$$

where $\tilde{\alpha}_1 = \alpha_1$, $\tilde{\alpha}_k = \alpha_k - \alpha_1 * \lfloor \alpha_k / \alpha_1 \rfloor$ for $2 \le k \le t$ and

$\tilde{\beta}(i, j) = \beta(i, j) - \alpha_1 * \lfloor \beta(i, j)/\alpha_1 \rfloor$ for $1 \le i < j \le d$.

Note that at least one of the $\tilde{\alpha}_k$'s , $k \ge 2$, or $\tilde{\beta}(i, j)$'s does not equal zero. Otherwise there would have been an element of order α_1 in $I(G')$, which is a contradiction to the fact that G is torsionfree.

7. Return to 2.

8. If $\alpha_1 = 1$, use the relation to eliminate b_1 from the set of generators of $I(G')$. If there are no more relations, go to 9. Else, return to 2 with a system of relations having one less generator and one less relation.

9. Introduce s elements into the presentation corresponding to the s independent generators remaining after solving the system of relations. The c_{ij}'s , $1 \le i < j \le d$ can be expressed in terms of these generators. Set $[a_j, a_i]$ to be the corresponding word in the s elements. This gives a canonical presentation.

This process must terminate since on each passage through step 2, either the number of relations decreases, or the magnitude of the smallest non-zero coefficient in the system is reduced.

An example is given.

Let G be presented by

$$\left\langle a_1, a_2, a_3, b; b^2 = [a_2, a_1][a_3, a_2]^4, [a_3, a_1]^2[a_3, a_2]^3 = \emptyset, \text{ class } 2 \right\rangle .$$

The relations become

$$2b - c_{12} - 4c_{23} = 0 , \tag{3}$$

$$2c_{13} - 3c_{23} = 0 . \tag{4}$$

Relation (3) is used to eliminate c_{12} . The system of relations remaining is just relation (4).

The minimum remaining coefficient is 2 . Let $\bar{c}_{13} = c_{13} - c_{23}$. Then

$$c_{13} = \bar{c}_{13} + c_{23} . \tag{5}$$

Relation (4) becomes

$$2\bar{c}_{13} - c_{23} = 0 . \tag{6}$$

Use this to eliminate c_{23} . Thus there are 2 remaining independent generators of $I(G')$, namely b and $\bar{c}_{13}$. In terms of b and $\bar{c}_{13}$, we have $c_{23} = 2\bar{c}_{13}$, $c_{13} = 3\bar{c}_{13}$ from (5) and $c_{12} = 2b - 8\bar{c}_{13}$ from (3). A canonical presentation for G is then

$$\langle a_1, a_2, a_3, b_1, b_2; [a_2, a_1] = b_1^2 b_2^{-8}, [a_3, a_1] = b_2^3,$$
$$[a_3, a_2] = b_2^2, \text{ canonical} \rangle .$$

REFERENCES

Gilbert Baumslag (1971), *Lecture Notes on Nilpotent Groups* (Conference Board of the Mathematical Sciences, Regional Conference Series in Mathematics, 2. American Mathematical Society, Providence, Rhode Island).

Gilbert Baumslag (1974), "Residually finite groups with the same finite images", *Compositio Math.* 29, 249-252.

I. Borosh and A.S. Fraenkel (1966), "Exact solutions of linear equations with rational coefficients by congruence techniques", *Math. Comp.* 20, 107-112.

N. Bourbaki (1959), *Elements de Mathématique*, XXIV. Livre II, *Algèbre*. Chapitre IX, *Formes Sesquilinéaires et Formes Quadratiques* (Hermann, Paris).

Gordon H. Bradley (1971), "Algorithms for Hermite and Smith normal matrices and linear Diophantine equations", *Math. Comp.* 25, 897-907.

H.R. Brahana (1940), "Finite metabelian groups and Plücker line-Coördinates", *Amer. J. Math.* 62, 365-379.

Richard P. Brent (1978), "Algorithm 524.MP, a Fortran multiple-precision arithmetic package [A1]", *ACM Trans. Math. Software* 4, 71-81.

Stanley Cabay (1971), "Exact solution of linear equations", *Second Symposium on Symbolic and Algebraic Manipulation*, 392-398 (Proc. Sympos., Los Angeles, 1971. ACM, New York).

S. Cabay and T.P. Lam (1977), "Congruence techniques for the exact solution of integer systems of linear equations", *ACM Trans. Math. Software* 3, 386-397.

H. Davenport (1970), *The Higher Arithmetic*, Fourth edition (Hutchinson University Library, London).

Leonard Eugene Dickson (1939), *Modern Elementary Theory of Numbers* (University of Chicago Press, Chicago and London).

R. Fricke and F. Klein (1897), *Vorlesungen über die Theorie der Automorphen Funktionen*, Volume 1 (Teubner, Leipzig. Reprinted by Johnson Reprint, New York, 1965).

L.E. Fuller (1955), "A canonical set of matrices over a principal ideal ring modulo m ", *Canad. J. Math.* 7, 54-59.

Michael A. Gauger (1973), "On the classification of metabelian Lie algebras", *Trans. Amer. Math. Soc.* 179, 293-329.

Michael A. Gauger (1974), "Duality theory for metabelian Lie algebras", *Trans. Amer. Math. Soc.* 187, 89-102.

Larry J. Gerstein (1977), "A local approach to matrix equivalence", *Linear Algebra Appl.* 16, 221-232.

Fritz Grunewald (1980), Private Communication.

Fritz J. Grunewald and Rudolf Scharlau (1979), "A note on finitely generated torsion-free nilpotent groups of class 2 ", *J. Algebra* 58, 162-175.

Fritz J. Grunewald and Daniel Segal (1979a), "The solubility of certain decision problems in arithmetic and algebra", *Bull. Amer. Math. Soc.* 16, 915-918.

Fritz Grunewald and Daniel Segal (1979b), "Some general algorithms. I. Arithmetic groups. II. Nilpotent groups", preprint.

Marshall Hall, Jr. (1959), *The Theory of Groups* (MacMillan, New York).

Philip Hall (1969), *Nilpotent Groups* (Notes of lectures given at the Canadian Math. Congress, University of Alberta, August 1957. Reissued by Mathematics Department, Queen Mary College, London).

G.H. Hardy and E.M. Wright (1960), *An Introduction to the Theory of Numbers* Fourth edition (Oxford University Press, Oxford).

B. Hartley and T.O. Hawkes (1970), *Rings, Modules and Linear Algebra* (Chapman and Hall, London, Colchester).

George Havas (1974), "A Reidemeister-Schreier program", *Proc. Second Internat. Conf. Theory of Groups*, Australian National University, Canberra, 1973, 347-356 (Lecture Notes in Mathematics, 372. Springer-Verlag, Berlin, Heidelberg, New York).

George Havas and L.G. Kovács, "Distinguishing two 11-crossing knots". Talk given at the 1979 May meeting of the Australian Mathematical Society. Paper in preparation.

George Havas and M.F. Newman (1980), "Application of computers to questions like those of Burnside", *Burnside Groups Proceedings*, Bielefeld, Germany, 1977, 211-230 (Lecture Notes in Mathematics, 806. Springer-Verlag, Berlin, Heidelberg, New York).

George Havas and Tim Nicholson (1976), "Collection", *SYMSAC* '76, 9-14 (ACM Sympos. Symbolic and Algebraic Computation. Association for Computing Machinery, Yorktown Heights, New York).

George Havas, J.S. Richardson and Leon S. Sterling (1979), "The last of the Fibonacci groups", *Proc. Roy. Soc. Edinburgh A* 83, 199-203.

George Havas and Leon S. Sterling (1979), "Integer matrices and abelian groups", *EUROSAM* '79, 431-451 (Internat. Sympos. Symbolic and Algebraic Manipulation, Marseille, 1979. Lecture Notes in Computer Science, 72. Springer-Verlag, Berlin, Heidelberg, New York).

Burton W. Jones (1950), *The Arithmetic Theory of Quadratic Forms* (The Carus Mathematical Monographs, 10. Mathematical Association of America, New York).

Ravindran Kannan and Achim Bachem (1979), "Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix", *Siam J. Computing* 8, 499-507.

Donald E. Knuth (1969), *The Art of Computer Programming*. Volume 2: *Semi-numerical Algorithms* (Addison-Wesley, Reading, Menlo Park, London).

William Judson Le Veque (1956), *Topics in Number Theory*, Volume 1 (Addison-Wesley, Reading, Massachusetts).

C.C. Macduffee, "Matrices with elements in a principal ideal ring", *Bull. Amer. Math. Soc.* 39 (1933), 564-584.

Wilhelm Magnus, Abraham Karrass and Donald Solitar (1976), *Combinatorial Group Theory*, 2nd revised edition (Dover, New York).

M.L. Mehta (1977), *Elements of Matrix Theory* (Hindustan Publishing Corporation, Delhi).

Erwin Mrowka (1964), "Reduktion von positiven binären quadratischen Formen modulo S ", *Wiss. Z. Hochsch. Verkehrswesen 'Friedrich List' Dresden* 11, 135-138.

Thomas Muir (1911), "The theory of determinants", *Historical Order of Development*, Volume 2 (1911) (reprinted Dover, New York, 1960).

Trygve Nagell (1964), *Introduction to Number Theory*, second edition (Chelsea, New York).

M.F. Newman (1976a), "Calculating presentations for certain kinds of quotient groups", *SYMSAC* '76, 2-8 (ACM Sympos. Symbolic and Algebraic Computation. ACM, Yorktown Heights, New York).

M.F. Newman (1976b), "A computer aided study of a group defined by fourth powers", *Bull. Austral. Math. Soc.* 14, 293-297.

M.F. Newman (1977), "Determination of groups of prime-power order", *Group Theory*, Canberra 1975, 73-84 (Proc. Mini Conf. Australian National University, 1975. Lecture Notes in Mathematics, 573. Springer-Verlag, Berlin, Heidelberg, New York).

Morris Newman (1972), *Integral Matrices* (Academic Press, New York).

P.F. Pickel (1971), "Finitely generated nilpotent groups with isomorphic finite quotients", *Trans. Amer. Math. Soc.* 160, 327-341.

В.Н. Ремесленников [V.N. Remeslennikov] (1967), "Сопряженность подгрупп в нильпотентных группах" [Conjugacy of subgroups in nilpotent groups], *Algebra i Logika Sem.* 6, 61-76.

Pierre Samuel, "About Euclidean rings", *J. Algebra* 19, 282-301.

John Scheuneman (1967), "Two-step nilpotent Lie algebras", *J. Algebra* 7, 152-159.

David A. Smith (1966), "A basis algorithm for finitely generated abelian groups", *Math. Algorithms* 1, 13-26.

H.J.S. Smith (1861), "On systems of linear indeterminate equations and congruences", *Philos. Trans. Roy. Soc. London* cli, 293-326. See also: *The Collected Mathematical Papers of Henry John Stephen Smith*, Volume I, 367-409 (Chelsea, Bronx, New York, 1965).

```
C
C COPYRIGHT
C
C  THIS PROGRAM DRIVES A COLLECTION OF SUBROUTINES WHICH COMPUTE
C   A CANONICAL FORM FOR AN INTEGER MATRIX.
C   DESCRIPTIONS OF THE PROGRAM AND RESULTS OBTAINED WITH IT APPEAR IN
C   G.HAVAS AND L.S.STERLING, 'INTEGER MATRICES AND ABELIAN GROUPS'
C       PROC. 1979 EUROSAM CONFERENCE
C
C
C   THE LABELLED COMMON BLOCKS ARE ALL INITIALISED IN BLOCK DATA.
C   THEIR PURPOSE IS AS FOLLOWS.
C
C    MP - THIS CONTAINS PARAMETERS NECESSARY FOR THE RUNNING OF THE
C         MULTIPLE PRECISION PACKAGE,MP.
C
C    FORM - THIS CONTAINS VARIOUS FORMAT PARAMETERS.
C           THE ARRAY FMT CONTAINS THE INPUT FORMAT FOR THE RELATION
C           MATRIX. (DEFAULT IS (40I3))
C           IN,OUT ARE THE STANDARD INPUT AND OUTPUT UNITS.
C           INMAT IS THE FILE FROM WHERE THE MATRIX IS TO BE READ.
C             (DEFAULT = 10)
C           INBIN=-INMAT, IS THE INPUT UNIT WHEN THE MATRIX IS
C                         IN BINARY FORM.
C           OUTLEV IS AN INTEGER CONTROLLING THE LEVEL OF OUTPUT.
C            (DEFAULT =1, THE LEAST AMOUNT OF OUTPUT)
C           IDENT IS AN ARRAY CONTAINING AN IDENTIFIER FOR THE MATRIX.
C
C    DET - DET CONTAINS PARAMETERS NEEDED FOR DETERMINANT CALCULATIONS.
C
C    PRIM - PRIM CONTAINS ARRAYS FOR DETERMINANT CALCULATIONS.
C
C    IMDPAR - IMDPAR CONTAINS PARAMETERS USED FOR DIAGONALISATION.
C
C  THE CURRENT OPTIONS AVAILABLE ARE THE FOLLOWING.
C   THEY ARE STORED IN VARIABLES OPT--, WHERE THE LAST 2 LETTERS
C   GIVE A MNEMONIC CODE.
C
C   ID - INTEGER DIAGONALISATION.
C
C   MD - MODULAR DIAGONALISATION.
C
C   DC - CALCULATE NON-ZERO RXR DETERMINANTS OF THE MATRIX,
C        WHERE R IS THE MATRIX RANK. (SEE SUBROUTINE DETCAL)
C
C   PM - PRINT OUT THE MATRIX.
C
C   OL - CHANGE THE OUTPUT LEVEL
C
C   CI - CHANGE THE MATRIX INPUT FILE NUMBER.
C
C   GP - GENERATE PRIMES.
C
C   EX - EXIT
C
```

```
C  OTHER VARIABLES USED ARE
C   SPACE, WHICH GIVES THE SIZE OF Y,
C   OPTION, WHICH CONTAINS THE INPUT OPTION,
C   PARAM, A GENERAL PURPOSE INPUT PARAMETER,
C   EARLY, A LOGICAL FLAG NEEDED IN DETERMINANT CALCULATIONS,
C   FRONTY, THE FIRST USABLE LOCATION OF Y,
C   BATCH, A  LOGICAL FLAG CONTROLLING THE AMOUNT OF INFORMATION
C          ECHOED BY THE PROGRAM.
C   NEWLIT,NEWBOU,NEWPOW,NEWINC - NEW VALUES FOR PARAMETERS OF IMDPAR.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION IDENT(3),FMT(3),DETS(10)
      LOGICAL EARLY,NOTFCR,BATCH
      COMMON Y(11000)
      COMMON /MP/ B,T,M,LUN,MXR,R(500)
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT
      COMMON /DET/ DETS,NUMDET,NACTRO,NACTCL,DETMAT,EARLY,NOTFCR
      COMMON /PRIM/ PRIMES(50),IMODS(50),LENPRI
      COMMON /IMDPAR/ LITROW,BOUND,POWER,INCR,MAXINT
      DATA OPTID,OPTMD,OPTDC,OPTPM,OPTOL/2HID,2HMD,2HDC,2HPM,2HOL/
      DATA OPTCI,OPTGP,OPTEX/2HCI,2HGP,2HEX/
      SPACE=11000
      FRONTY=1
      LENPRI=IFIX(ALOG10(FLOAT(PRIMES(1))))+1
      BATCH=.FALSE.
      WRITE(OUT,205)
C
C  READ IN OPTION
C
10    WRITE(OUT,210)
      READ(IN,160)OPTION,PARAM
      IF(BATCH)WRITE(OUT,310)OPTION,PARAM
      IF(OPTION.EQ.OPTID)GO TO 15
      IF(OPTION.EQ.OPTMD)GO TO 35
      IF(OPTION.EQ.OPTDC)GO TO 55
      IF(OPTION.EQ.OPTPM)GO TO 85
      IF(OPTION.EQ.OPTOL)GO TO 100
      IF(OPTION.EQ.OPTCI)GO TO 120
      IF(OPTION.EQ.OPTGP)GO TO 130
      IF(OPTION.EQ.OPTEX)STOP
C
C   INCORRECT OPTION
C
      WRITE(OUT,220)OPTION
      GO TO 10
C
C   INTEGER MATRIX DIAGONALISATION.   (SEE SUBROUTINE IMDIAG)
C
15    IF(INMAT.GT.0)GO TO 20
      REWIND INBIN
      READ(INBIN,170)NREL,NCOL,IDENT
      GO TO 22
20    IF(INMAT.NE.IN)REWIND INMAT
      READ(INMAT,170)NREL,NCOL,IDENT,FMT
22    WRITE(OUT,240)IDENT
```

```
      IF((NREL+1)*NCOL.LE.SPACE)GO TO 30
25    WRITE(OUT,250)SPACE
      GO TO 10
30    CALL RUNIMD(Y(FRONTY),NREL+1,NCOL,PARAM)
      GO TO 10
C
C  MODULAR DIAGONALISATION.   (SEE MODIAG)
C   N.B. MODULO,THE NUMBER MODULO WHICH YOU CALCULATE,
C        MUST BE A POWER OF A PRIME.
C
35    IF(INMAT.GT.0)GO TO 40
      REWIND INBIN
      READ(INBIN,170)NREL,NCOL,IDENT
      GO TO 42
40    IF(INMAT.NE.IN)REWIND INMAT
      READ(INMAT,170)NREL,NCOL,IDENT,FMT
42    IF(PARAM.LE.0)GO TO 45
      MODULO=PARAM
      PARAM=0
      GO TO 50
45    WRITE(OUT,260)
      READ(IN,180)MODULO
50    WRITE(OUT,270)IDENT,MODULO
      IF((NREL+1)*NCOL.GT.SPACE)GO TO 25
      CALL MODIAG(Y(FRONTY),NREL+1,NCOL,MODULO,PARAM)
      GO TO 10
C
C   DETERMINANT CALCULATION.   (SEE DETCAL)
C    PARAM GIVES THE NUMBER OF DETERMINANTS TO BE CALCULATED.
C
55    IF(INMAT.GT.0)GO TO 58
      REWIND INBIN
      READ(INBIN,170)NREL,NCOL,IDENT
      GO TO 60
58    IF(INMAT.NE.IN)REWIND INMAT
      READ(INMAT,170)NREL,NCOL,IDENT,FMT
60    EARLY=PARAM.LT.0
      NUMDET=MAXO(MINO(10,IABS(PARAM)),1)
      WRITE(OUT,280)IDENT
      IF(NREL.LT.NCOL)GO TO 72
      ROWS=NCOL+NUMDET+1
      NOTFCR=.FALSE.
      GO TO 75
72    NOTFCR=.TRUE.
      ROWS=NREL+2
75    IF(NCOL*ROWS.GT.SPACE)GO TO 25
80    CALL DETCAL(Y(FRONTY),ROWS,NCOL,NREL)
      GO TO 10
C
C  PRINT OUT THE MATRIX
C
85    IF(INMAT.GT.0)GO TO 90
      REWIND INBIN
      READ(INBIN,170)NREL,NCOL,IDENT
      GO TO 95
```

```
90    IF(INMAT.NE.IN)REWIND INMAT
      READ(INMAT,170)NREL,NCOL,IDENT,FMT
95    WRITE(OUT,285)IDENT
      CALL MATOUT(Y(1),NREL,NCOL)
      GO TO 10
C
C  CHANGE THE OUTPUT LEVEL (PARAM CONTAINS THE NEW LEVEL)
C    A NEGATIVE OUTPUT LEVEL TURNS ON BATCH
C
100   BATCH=PARAM.LT.0
      OUTLEV=IABS(PARAM)
      GO TO 10
C
C  CHNGE THE INPUT FILE NUMBER (TO BE FOUND IN PARAM)
C
120   INMAT=PARAM
      INBIN=-INMAT
      GO TO 10
C
C  GENERATE PRIMES FOR A MACHINE WITH A WORDSIZE OF NBITS,
C       WHERE NBITS = PARAM.
C
130   IF(PARAM.GE.18)GO TO 135
      WRITE(OUT,350)PARAM
      GO TO 10
135   NBITS=PARAM
      J=0
      J=2*(2**(NBITS-2)-1)+1
      J=SQRT(FLOAT(J))
      J=J-MOD(J+1,2)+2
      K=SQRT(FLOAT(J))
      DO 500 I=1,50
430   J=J-2
      DO 450 L=3,K,2
      IF(MOD(J,L).EQ.0) GO TO 430
450   CONTINUE
      PRIMES(51-I)=J
500   CONTINUE
      IMODS(1)=1
      DO 550 I=2,50
      P=PRIMES(I)
      TEMP=1
      II=I-1
      DO 520 J=1,II
520   TEMP=MOD(TEMP*PRIMES(J),P)
      CALL DIV(TEMP,P,DUMMY,INV)
550   IMODS(I)=INV
      IF(OUTLEV.NE.17)GO TO 10
      WRITE(OUT,340)(PRIMES(I),I=1,50)
      WRITE(OUT,340)(IMODS(I),I=1,50)
      GO TO 10
C
160   FORMAT(A2,I3)
170   FORMAT(2I4,6A4)
180   FORMAT(I10)
```

```
200   FORMAT(3I3,I6,I12)
C
205   FORMAT(44H INTEGER MATRIX DIAGONALISATION PROGRAM V2.1)
210   FORMAT(14H OPTION(A2,I3))
220   FORMAT(15H ILLEGAL OPTION,5X,A2)
240   FORMAT(20HODIAGONALISATION OF ,3A4)
250   FORMAT(36H THE ARRAY SIZE EXCEEDS THE LIMIT OF,I8,6H WORDS)
260   FORMAT(18H INPUT MODULO(I10))
270   FORMAT(20HODIAGONALISATION OF ,3A4,7H MODULO,I12)
280   FORMAT(29HODETERMINANT CALCULATION FOR ,3A4)
285   FORMAT(11H MATRIX OF ,3A4)
290   FORMAT(15H INPUT FMT(3A4))
310   FORMAT(3X,A2,3X,I3)
320   FORMAT(2X,I5,2X,I5)
330   FORMAT(2X,3A4)
340   FORMAT(1H ,8(I7,1H,))
350   FORMAT(12H WORDSIZE OF,I3,29H BITS IS TOO SMALL FOR PRIME ,
     *      10HGENERATION)
      END
      SUBROUTINE IMDIAG (A,NROW,NCOL,TOP,BOTTOM,NORED)
C
C  THIS SUBROUTINE CONVERTS AN INTEGER MATRIX INTO A DIAGONAL FORM,
C   BETWEEN ROWS TOP AND BOTTOM.
C   THE MATRIX WITH WHICH WE ARE WORKING, IS FOUND IN ARRAY A,
C   WHICH HAS DIMENSIONS NROW BY NCOL. THE LAST ROW OF A CONTAINS
C   COLUMN NAMES.
C
C   VARIABLES USED BY THIS ROUTINE ARE
C  CASES - THE DIMENSION OF THE LARGEST SQUARE SUBMATRIX OF A.
C  MINVAL - THIS KEEPS TRACK OF THE MINIMUM ABSOLUTE VALUE OF A NON-ZERO
C           ELEMENT OF A.
C  MINI,MINJ - THESE MARK THE POSITION OF THE ELEMENT WITH ABSOLUTE
C              VALUE MINVAL.
C  MAXINT - THE LARGEST MACHINE-REPRESENTABLE INTEGER.
C      STORED IN COMMON BLOCK IMDPAR, ITS VALUE IS SET IN BLOCK DATA.
C   THE OTHER PARAMETERS OF IMDPAR ARE DESCRIBED IN THE USERS GUIDE.
C    OTHER VARIABLES CONTROLLING THE INVOCATION OF THE REDUCTION
C    ROUTINE REDROW ARE
C  NORED - PREVENTS REDROW BEING CALLED, RETURNING CONTROL TO RUNIMD.
C  MAXELT - STORES THE LARGEST ABSOLUTE VALUE OF AN ENTRY OF A,
C           IF IT IS LARGER THAN 2**POWER.
C  MAXCOL - THE COLUMN IN WHICH MAXELT OCCURS.
C    AT EACH ITERATION MAXELT IS SET EQUAL TO 2**POWER2. IF MAXELT
C    BECOMES LARGER REDROW IS CALLED.
C
C  ARRAY WSPACE IN COMMON BLOCK MP IS USED TO HELP IN PRINTOUT.
C
      IMPLICIT INTEGER (A-Z)
      LOGICAL NORED
      DIMENSION A(NROW,NCOL),FMT(3)
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT(3)
      COMMON /IMDPAR/LITROW,BOUND,POWER,INCR,MAXINT
      COMMON /MP/ B,T,M,LUN,MXR,WSPACE(500)
C
      NREL=NROW-1
```

```
      CASES=MINO(NREL,NCOL)
      BOTTOM=MINO(CASES,BOTTOM)
      POWER2=POWER
      MAXELT=2**POWER2
C
      DO 100 ITER=TOP,BOTTOM
C
C  CHECK MAXELT TO SEE IF REDUCTION IS NECESSARY.
C   IF SO CALL REDROW.
C
      IF (MAXELT.EQ.2**POWER2) GO TO 20
9     IF(.NOT.NORED)GO TO 10
      TOP=ITER
      RETURN
10    IF (POWER2.LT.BOUND) GO TO 15
      WRITE (OUT,120)
      RETURN
15    IF (OUTLEV.GE.2) WRITE (OUT,125) MAXELT,POWER2
      CALL REDROW (A,NROW,NCOL,ITER,MAXCOL)
      POWER2=POWER2+INCR
      MAXELT=2**POWER2
C
C  SEARCH FOR MINIMAL ARRAY ELEMENT, ESPECIALLY A 1.
C
20    MINVAL=MAXINT
      MINSUM=MAXINT
      DO 40 I=ITER,NREL
      DO 25 J=ITER,NCOL
      IF (A(I,J).EQ.0) GO TO 25
      IF (IABS(A(I,J)).EQ.1) GO TO 30
      IF (IABS(A(I,J)).GE.MINVAL) GO TO 25
      MINVAL=IABS(A(I,J))
      MINI=I
      MINJ=J
25    CONTINUE
      GO TO 40
C
C  A 1 HAS BEEN FOUND. CALCULATE ITS ROWSUM.
C  CHECK IF IT IS BETTER THAN PREVIOUS VALUES, OR GOOD ENOUGH
C  TO BE USED IMMEDIATELY.
C
30    ROWSUM=0
      DO 35 K=ITER,NCOL
35    ROWSUM=ROWSUM+IABS(A(I,K))
      IF (ROWSUM.GE.MINSUM) GO TO 40
      MINVAL=1
      MINSUM=ROWSUM
      MINI=I
      MINJ=J
      IF (ROWSUM.LE.LITROW) GO TO 70
40    CONTINUE
      IF (MINVAL.EQ.1) GO TO 70
C
C  IF MINVAL STILL EQUALS MAXINT, THERE ARE ONLY ZEROS IN THE MATRIX.
C
```

```
      IF (MINVAL.EQ.MAXINT) GO TO 102
C
C  CHECK THAT THE MINIMAL ELEMENT DIVIDES THE ENTRIES IN ITS COLUMN
C
45    DO 55 I=ITER,NREL
      IF (MOD(A(I,MINJ),MINVAL).EQ.0) GO TO 55
C
C  DIVISIBILITY CONDITIONS ARE NOT MET. GENERATE A NEW MINIMAL ELEMENT,
C   BY PERFORMING THE APPROPRIATE ELEMENTARY ROW OPERATION.
C  WHEN SUBTRACTING (OR ADDING) ROWS WATCH FOR BIG ELEMENTS.
C
      KILFAC=A(I,MINJ)/A(MINI,MINJ)
      IF(IABS(A(I,MINJ))-IABS(KILFAC)*MINVAL.GT.MINVAL/2)KILFAC=
     1     ISIGN(IABS(KILFAC)+1,KILFAC)
      DO 50 J=ITER,NCOL
      A(I,J)=A(I,J)-KILFAC*A(MINI,J)
      IF(A(I,J).EQ.0)GO TO 50
      IF(IABS(A(I,J)).LT.MINVAL)GO TO 48
      IF (IABS(A(I,J)).LE.MAXELT) GO TO 50
      MAXELT=IABS(A(I,J))
      MAXCOL=J
      GO TO 50
48    MINVAL=IABS(A(I,J))
      MINJ=J
50    CONTINUE
      IF (MAXELT.NE.2**POWER2) GO TO 9
C
C  CHECK NEW MINIMAL ELEMENT. IF A 1, GO STRAIHT TO KILLING OFF.
C   IF NOT, GO BACK AND CHECK DIVISIBILITY CONDITIONS.
C
      MINI=I
      IF (MINVAL.EQ.1) GO TO 70
      GO TO 45
55    CONTINUE
C
C  NOW CHECK IF THE MINIMAL ELEMENT DIVIDES ENTRIES IN ITS ROW.
C
      DO 65 J=ITER,NCOL
      IF (MOD(A(MINI,J),MINVAL).EQ.0) GO TO 65
C
C  THIS TIME USE COLUMN OPERATIONS TO GENERATE A NEW MINIMAL ELEMENT,
C   PROCEEDING IN THE ANALOGOUS WAY TO BEFORE
C
      KILFAC=A(MINI,J)/A(MINI,MINJ)
      IF(IABS(A(MINI,J))-IABS(KILFAC)*MINVAL.GT.MINVAL/2)
     1   KILFAC=ISIGN(IABS(KILFAC)+1,KILFAC)
      DO 60 I=ITER,NREL
      A(I,J)=A(I,J)-KILFAC*A(I,MINJ)
      IF(A(I,J).EQ.0)GO TO 60
      IF(IABS(A(I,J)).LT.MINVAL)GO TO 58
      IF (IABS(A(I,J)).LE.MAXELT) GO TO 60
      MAXELT=IABS(A(I,J))
      MAXCOL=J
      GO TO 60
58    MINVAL=IABS(A(I,J))
```

```
      MINI=I
60    CONTINUE
      IF(OUTLEV.GE.4)WRITE(OUT,130)A(NROW,J),A(NROW,J),KILFAC,
     1            A(NROW,MINJ)
      IF (MAXELT.NE.2**POWER2) GO TO 9
C
      MINJ=J
      IF (MINVAL.EQ.1) GO TO 70
      GO TO 45
65    CONTINUE
C
C  THE MINIMAL ELEMENT WHICH NOW DIVIDES ENTRIES IN ITS ROW AND COLUMN
C   IS BROUGHT TO THE TOP LEFT-HAND CORNER OF THE SUBMATRIX
C   BEING PROCESSED.
C
70    IF (MINI.NE.ITER) CALL ROWSWP (A,NROW,NCOL,ITER,MINI)
      IF (MINJ.NE.ITER) CALL COLSWP (A,NROW,NCOL,ITER,MINJ)
      IF (ITER.EQ.CASES) GO TO 85
C
C  KILL OFF THE ENTRIES IN THE LEFTHAND COLUMN UNDER THE MINIMAL ELEMENT
C
74    START=ITER+1
      DO 80 I=START,NREL
      IF(A(I,ITER).EQ.0)GO TO 80
      KILFAC=A(I,ITER)/A(ITER,ITER)
      DO 75 J=START,NCOL
      A(I,J)=A(I,J)-A(ITER,J)*KILFAC
      IF (IABS(A(I,J)).LE.MAXELT) GO TO 75
      MAXELT=IABS(A(I,J))
      MAXCOL=J
75    CONTINUE
80    CONTINUE
C
C  IF OUTPUT OPTION IS SET, PRINT VALUE OF ELIMINATED COLUMN.
C
85    IF (OUTLEV.LT.3) GO TO 100
      DIAGEL=-A(ITER,ITER)
      LENGTH=0
      IF (ITER.EQ.CASES) GO TO 95
      DO 90 J=START,NCOL
      IF (A(ITER,J).EQ.0) GO TO 90
      LENGTH=LENGTH+2
      WSPACE(LENGTH)=A(NROW,J)
      WSPACE(LENGTH-1)=A(ITER,J)
90    CONTINUE
      IF (LENGTH.EQ.0) GO TO 95
      WRITE (OUT,135) DIAGEL,A(NROW,ITER),(WSPACE(I),I=1,LENGTH)
      GO TO 100
95    WRITE (OUT,140) DIAGEL,A(NROW,ITER)
100   CONTINUE
C
C  OUTPUT STRUCTURE OF DIAGONALISED MATRIX.
C
102   IF(BOTTOM.NE.CASES)RETURN
      NONTRV=0
```

```
      DO 110 K=1,CASES
      IF(IABS(A(K,K))-1)115,110,105
105   NONTRV=NONTRV+1
      WSPACE(NONTRV)=IABS(A(K,K))
110   CONTINUE
      K=CASES+1
115   RANK=K-1
      WRITE(OUT,145)NREL,NCOL,RANK
      IF(NONTRV.EQ.0)WRITE(OUT,150)
      IF(NONTRV.GT.0)WRITE(OUT,160)(WSPACE(I),I=1,NONTRV)
      RETURN
C
C
120   FORMAT (48HOSTOP DIAGONALISATION SINCE OVERFLOW VERY LIKELY)
125   FORMAT (8HOELEMENT,I12,16H LARGER THAN 2**,I2,9H DETECTED)
130   FORMAT(19H COLUMN REPLACEMENT,I6,2H =,4X,3H1 *,I6,I8,2H *,I6)
135   FORMAT (I7,2H *,I5,3H = ,8(I6,2H *,I5)/(17X,8(I6,2H *,I5)))
140   FORMAT (I7,2H *,I5,2H =,6X,1HO)
145   FORMAT(17HONUMBER OF ROWS =,I12/2OH NUMBER OF COLUMNS =,I9/
     1        7HORANK =,I10)
150   FORMAT (42HOTHERE ARE NO NONTRIVIAL DIAGONAL ELEMENTS)
160   FORMAT(37HOTHE NONTRIVIAL DIAGONAL ELEMENTS ARE,15I6/(37X,15I6))
      END
      BLOCK DATA
C
C  BLOCK DATA CONTAINS MANY OF THE MACHINE DEPENDENT CONSTANTS.
C  THIS VERSION IS SUITED FOR A MACHINE WITH 32 BIT INTEGER WORDSIZE.
C  THE VALUES ARE ALSO CORRECT, THOUGH NOT OPTIMAL, FOR A MACHINE WITH
C  LARGER WORDSIZE. HOWEVER FOR A SMALLER MACHINE ALTERATIONS ARE
C  NECESSARY. VARIABLES TO BE CHANGED ARE:-
C  NBITS - THE BIT SIZE FOR AN INTEGER.
C  ARRAYS PRIMES AND IMODS. THESE ARE A LIST OF OPTIMAL PRIMES USED
C  FOR DETERMINANT CALCULATIONS. THE OPTIMAL PRIMES ARE THE 50 LARGEST
C  PRIMES SMALLER THAN THE SQUARE ROOT OF THE MACHINE WORDSIZE.
C  ELEMENTS OF IMODS ARE CALCULATED SO THAT
C  IMODS(I)*PRIMES(1)*...*PRIMES(I-1) = 1 (MOD PRIMES(I))
C  PARAMETERS OF COMMON BLOCK MP. CORRECT VALUES CAN BE DETERMINED
C  FROM MP USERS GUIDE.
C
C  DEFAULT UNITS ARE:-
C   5 FOR THE INPUT UNIT IN.
C   6 FOR THE OUTPUT UNIT OUT.
C   12 FOR DETMAT, A TEMPORARY UNIT FOR READING AND WRITING A MATRIX
C                  DURING DETERMINANT CALCULATIONS.
C   10 FOR INMAT, THE MATRIX INPUT UNIT.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION FMT(3),DETS(10)
      LOGICAL EARLY,NOTFCR
      COMMON /MP/B,T,M,LUN,MXR,R(500)
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT(3)
      COMMON /IMDPAR/ LITROW,BOUND,POWER2,INCR,MAXINT
      COMMON /DET/ DETS,NUMDET,NACTRO,NACTCL,DETMAT,EARLY,NOTFCR
      COMMON /PRIM/ PRIMES(50),IMODS(50),LENPRI
      DATA B,T,M,MXR,LUN/10000,60,100,500,6/
```

```
      DATA IN,OUT,INMAT,OUTLEV/5,6,10,1/
      DATA FMT(1),FMT(2)/4H(40I,4H3)  /
      DATA LITROW,BOUND,POWER2,INCR,NBITS/100,26,10,1,32/
      DATA DETMAT /12/
      DATA PRIMES /45779,45817,45821,45823,45827,45833,45841,45853,
     *45863,45869,45887,45893,45943,45949,45953,45959,45971,45979,
     *45989,46021,46027,46049,46051,46061,46073,46091,46093,46099,
     *46103,46133,46141,46147,46153,46171,46181,46183,46187,46199,
     *46219,46229,46237,46261,46271,46273,46279,46301,46307,46309,
     *46327,46337/
      DATA IMODS /1,37377,11728,23519,43110,30088,2526,16401,37649,
     *18578,27899,17890,21803,20804,10693,9334,25261,43378,31537,
     *24747,28760,44377,39985,8095,13495,14627,20635,7285,25732,
     *23043,28922,26658,5105,9024,44212,11688,26150,40002,38707,9376,
     *7784,6996,37221,22232,2444,40578,36280,25262,36985,14830/
      END
      SUBROUTINE COLSWP(A,M,N,K,L)
C
C  COLSWP SWAPS THE KTH AND LTH COLUMNS OF ARRAY A.
C  N.B. K IS ALWAYS LESS THAN L.
C       ALSO COLSWP IS CALLED WHEN WE ARE DEALING WITH THE SUBMATRIX
C       OF A WHOSE TOP LEFT-HAND ELEMENT IS A(K,K) AND THUS WE NEED
C       ONLY SWAP FROM THE KTH ROW ONWARDS.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(M,N)
      DO 100 I=K,M
      TEMP=A(I,K)
      A(I,K)=A(I,L)
100   A(I,L)=TEMP
      RETURN
      END
      SUBROUTINE DETCAL (A,NROW,NCOL,NREL)
C
C  THIS ROUTINE IS THE MAJOR PART OF THE DETERMINANT-CALCULATING
C  CODE COMPRISING SUBROUTINES DETCAL,DETOUT,FDRANK AND GFPDET.
C  COLLECTIVELY THE FOLLOWING IS DONE TO THE INTEGER MATRIX A.
C  FIRSTLY FDRANK FINDS A SQUARE SUBMATRIX OF A OF
C  MAXIMAL RANK. THE DETERMINANT OF THIS MATRIX IS CALCULATED MODULO
C  SEVERAL PRIMES BY GFPDET. USING THE CHINESE REMAINDER THEOREM,
C  THE DETERMINANT OF THE MATRIX IS RECONSTRUCTED.
C  THIS NUMBER MAY BE LARGER THAN THE INTEGER WORD SIZE, SO IT IS
C  STORED IN MIXED RADIX REPRESENTATION (SEE KNUTH). THE DETERMINANT
C  IS THEN CALCULATED AS A MULTIPLE PRECISION INTEGER IN DETOUT.
C  THE DETERMINANT IS CALCULATED TO DETERMINE THE
C  DIAGONAL ELEMENTS OF THE SMITH NORMAL FORM OF A. NOW THE
C  PRODUCT OF THESE ELEMENTS EQUALS THE GCD OF ALL SQUARE SUBMATRICES
C  OF MAXIMAL RANK.THUS WE CALCULATE SEVERAL DETERMINANTS SIMULTANEOUSLY
C  BY REPLACING THE LAST ROW (OR COLUMN) OF THE SUBMATRIX, TO GET
C  A BETTER APPROXIMATION TO THIS GCD.
C
C  NOTATION IN THESE SUBROUTINES HAS BEEN KEPT CONSISTENT WHERE
C  POSSIBLE.  VARIABLES AND ARRAYS USED ARE DESCRIBED BELOW.
C
C  COMMON BLOCK PRIM CONTAINS
```

```
C  PRIMES - THIS ARRAY STORES 50 PRIMES USED IN CALCULATING DETERMINANTS
C           THE PRIMES CHOSEN HERE ARE SUITABLE FOR A MACHINE
C           WITH A 32 BIT WORDSIZE.THEY CAN BE CHANGED IN BLOCKDATA.
C  IMODS - THIS ARRAY CONTAINS VARIOUS INVERSES NEEDED TO CALCULATE
C          COEFFICIENTS OF MIXED RADIX REPRESENTATIONS.
C  LENPRI - THIS IS THE LENGTH OF A PRIME, USED TO POSITION OUTPUT
C           IN DETOUT.
C
C  COMMON BLOCK DET CONTAINS
C  DETS - AN ARRAY CONTAINING DIFFERENT DETERMINANTS CALCULATED
C         IN GFPDET.
C  NUMDET - THE NUMBER OF DIFFERENT DETERMINANTS TO BE CALCULATED
C           SIMULTANEOUSLY.  (UP TO 10 POSSIBLE)
C  NACTRO - ORIGINALLY SET TO NREL, NACTRO IS CALCULATED IN FDRANK TO BE
C         THE NUMBER OF ROWS OF A BEING KEPT.
C  NACTCL - THE EFFECTIVE NUMBER OF ACTIVE COLUMNS.
C           N.B. EITHER NACTRO OR NACTCL REPRESENTS THE RANK OF A.
C  DETMAT - A FILE CONTAINING A COPY OF THE SUBMATRIX USED.
C           IT IS WRITTEN IN FDRANK AND USED IN GFPDET AND HERE.
C  EARLY - A LOGICAL FLAG DESCRIBED WHEN DISCUSSING THE BOUND.
C  NOTFCR - A LOGICAL FLAG. WHEN TRUE, IT MEANS THAT THE NUMBER OF
C           LINEARLY INDEPENDENT ROWS IS LESS THAN THE NUMBER OF COLUMNS
C           SO WE REPLACE THE LAST COLUMN RATHER THAN THE LAST ROW WHEN
C           CALCULATING VARIOUS DETERMINANTS.
C
C  A - THE INTEGER MATRIX BEING PROCESSED.
C  NCOL - THE NUMBER OF COLUMNS OF A.
C  NREL - THE NUMBER OF ROWS OF A.
C  NROW - THE ROW DIMENSION OF A DETERMINED BY THE MAIN PROGRAM.
C
C  AFTER FDRANK IS CALLED TO FIND THE REQUISITE SUBMATRIX,
C  ITS HADAMARD BOUND IS CALCULATED.
C  VARIABLE BOUND STORES THE LOG OF THIS BOUND. NUMPRM IS THEN THE
C  NUMBER OF PRIMES NEEDED TO BE SURE OF EXCEEDING THIS BOUND.
C  IN GENERAL THE DETERMINANT DOES NOT APPROACH ITS HADAMARD BOUND,
C  AND ONE MAY STOP AS SOON AS A ZERO RADIX COEFFICIENT IS FOUND.
C  EARLY IS SET IF YOU WANT TO STOP AT THIS EARLY POINT.
C  STOPCR CONTROLS WHEN YOU STOP CALCULATING DETERMINANTS.
C
C  RADIX - THIS ARRAY STORES THE VARIOUS MIXED RADIX COEFFICIENTS.
C          RADIX(I,J) IS THE ITH COEFFICIENT OF THE JTH DETERMINANT.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(NROW,NCOL),RADIX(50,10)
      DIMENSION FMT(3),DETS(10)
      REAL BOUND
      LOGICAL EARLY,NOTFCR,STOPCR
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT(3)
      COMMON /DET/ DETS,NUMDET,NACTRO,NACTCL,DETMAT,EARLY,NOTFCR
      COMMON /PRIM/ PRIMES(50),IMODS(50),LENPRI
C
C  FIND RANK OF MATRIX.  N.B. AFTER FDRANK IS CALLED, THE LAST ROW OF
C    DETMAT IS AN INDEX ROW, WHOSE USE IS DESCRIBED IN FDRANK.
C
      STOPCR=.FALSE.
```

```
      REWIND DETMAT
      NACTCL=NCOL
      NACTRO=NREL
      PRIME=PRIMES(50)
      CALL FDRANK (A,NROW,NCOL,PRIME)
      END FILE DETMAT
      IF (NACTRO.NE.0) GO TO 10
      WRITE (OUT,120)
      RETURN
C
10    IF((.NOT.HORIZ).OR.(NOTFCR.AND.(NACTRO.EQ.NREL)))GO TO 20
C
C  OPTIONAL RANK VERIFICATION HERE
C
C  CALCULATE HADAMARD BOUND FOR MATRIX.
C
20    ROWS=NCOL-1
      REWIND DETMAT
      BOUND=0.
      IF(NOTFCR)GO TO 200
C
C  MATRIX RANK = COLUMN RANK.
C
      DO 30 I=1,ROWS
      SUM=0
      READ(DETMAT)(A(1,J),J=1,NCOL)
      DO 25 J=1,NCOL
25    SUM=SUM+A(1,J)*A(1,J)
30    BOUND=BOUND+ALOG(FLOAT(SUM))
      MAXSUM=0
      DO 35 I=1,NUMDET
      SUM=0
      READ(DETMAT)(A(1,J),J=1,NCOL)
      DO 33 J=1,NCOL
33    SUM=SUM+A(1,J)*A(1,J)
35    IF(SUM.GT.MAXSUM)MAXSUM=SUM
      BOUND=BOUND+ALOG(FLOAT(MAXSUM))
      GO TO 40
C
C  MATRIX RANK IS LESS THAN COLUMN RANK.
C
200   ROWS=NACTRO+1
      DO 210 I=1,NACTRO
210   READ(DETMAT)(A(I,J),J=1,NCOL)
      READ(DETMAT)(A(ROWS,J),J=1,NACTCL)
      CASES=NACTRO-1
      DO 230 K=1,CASES
      J=A(ROWS,K)
      SUM=0
      DO 240 I=1,NACTRO
240   SUM=SUM+A(I,J)*A(I,J)
230   BOUND=BOUND+ALOG(FLOAT(SUM))
      MAXSUM=0
      DO 270 K=1,NUMDET
      SUM=0
```

```
      J=A(ROWS,CASES+K)
      DO 260 I=1,NACTRO
260   SUM=SUM+A(I,J)*A(I,J)
270   IF(SUM.GT.MAXSUM)MAXSUM=SUM
      BOUND=BOUND+ALOG(FLOAT(MAXSUM))
40    BOUND=BOUND/2.+ALOG(2.)
C
C  CALCULATE NO. OF PRIMES NEEDED
C
      NUMPRM=0
45    NUMPRM=NUMPRM+1
      BOUND=BOUND-ALOG(FLOAT(PRIMES(NUMPRM)))
      IF (BOUND.GT.0.) GO TO 45
C
C  HEADINGS
C
      IF(OUTLEV.GE.3)WRITE(OUT,140)NUMDET,NACTRO,NUMPRM
      IF(OUTLEV.GE.2)WRITE(OUT,170)
C
C  ITERATIVELY CALCULATE THE DETERMINANT
C
      DO 100 ITER=1,NUMPRM
      PRIME=PRIMES(ITER)
      CALL GFPDET (A,NROW,NCOL,PRIME)
      IF(NACTRO.EQ.0)GO TO 115
C
C  CONVERT MODULAR DETERMINANTS TO MIXED RADIX COEFFICIENTS
C
      RANGE=(PRIME-1)/2
      IF (ITER.NE.1) GO TO 60
      DO 50 I=1,NUMDET
      RADIX(1,I)=DETS(I)
      IF(RADIX(1,I).GT.RANGE)RADIX(1,I)=RADIX(1,I)-PRIME
50    IF (OUTLEV.GE.2) WRITE (OUT,160) ITER,PRIME,DETS(I),RADIX(1,I)
      GO TO 100
60    DO 90 I=1,NUMDET
      TEMP=RADIX(ITER-1,I)
      IF (ITER.EQ.2) GO TO 80
      DO 70 J=3,ITER
      K=ITER-J+1
70    TEMP=MOD(RADIX(K,I)+PRIMES(K)*TEMP,PRIME)
80    TEMP=MOD(MOD(DETS(I)-TEMP,PRIME)*IMODS(ITER),PRIME)
      IF(IABS(TEMP).GT.RANGE)TEMP=TEMP-ISIGN(PRIME,TEMP)
      RADIX(ITER,I)=TEMP
      IF(EARLY.AND.(TEMP.EQ.0))STOPCR=.TRUE.
90    IF (OUTLEV.GE.2) WRITE (OUT,160) ITER,PRIME,DETS(I),RADIX(ITER,I)
      IF(STOPCR)GO TO 110
100   CONTINUE
      ITER=NUMPRM
110   CALL DETOUT (RADIX,ITER,NUMDET)
      RETURN
C
115   WRITE(OUT,130)
      RETURN
C
```

```
120   FORMAT (16H MATRIX OF ZEROS)
130   FORMAT(47H THIS LIST OF PRIMES UNSUITABLE FOR THIS MATRIX)
140   FORMAT(I4,30H DETERMINANTS CALCULATED USING,I5,5H ROWS/8H AT MOST,
     1  I3,18H PRIMES ARE NEEDED)
160   FORMAT (I4,3X,I6,4X,I7,6X,I7)
170   FORMAT (40H ITER  PRIME  DETERMINANTS  RADIX COEFFS)
      END
      SUBROUTINE DETOUT (RADIX,ITER,NUMDET)
C
C  DETOUT RECONSTRUCTS NUMBERS FROM MIXED RADIX REPRESENTATIONS,
C  AND CALCULATES THEIR GCD. COLUMNS OF RADIX GIVE THE MIXED RADIX
C  COEFFICIENTS AND ARRAY PRIMES CONTAINS THE PRIMES.
C  THE NUMBER IS CALCULATED AS A MULTIPLE PRECISION INTEGER USING A
C  THE MP PACKAGE OF BRENT.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION RADIX(50,10),TEMP(100),LAST(100),ANS(100),FMT(3)
      COMMON /MP/ B,T,M,LUN,MXR,R(500)
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT(3)
      COMMON /PRIM/ PRIMES(50),IMODS(50),LENPRI
C
C  CHECK IF ITER = 1.
C
      IF (ITER.GT.1) GO TO 25
      GCD=IABS(RADIX(1,1))
      DO 20 I=1,NUMDET
      WRITE (OUT,60) I,RADIX(1,I)
      IF (I.EQ.1) GO TO 20
      VAL=IABS(RADIX(1,I))
      IF (VAL.GE.DETERM) GO TO 10
      TEM=VAL
      VAL=DETERM
      DETERM=TEM
10    CALL DIV (DETERM,VAL,GCD,DUMMY)
      WRITE (OUT,70) GCD
20    DETERM=GCD
      RETURN
C
25    START=MAX0(1,100-LENPRI*ITER)
      DO 50 I=1,NUMDET
      CALL MPCIM (RADIX(ITER,I),TEMP)
      DO 30 J=2,ITER
      K=ITER+1-J
      CALL MPMULI (TEMP,PRIMES(K),TEMP)
      CALL MPADDI (TEMP,RADIX(K,I),TEMP)
30    CONTINUE
      CALL MPOUT (TEMP,ANS,100,-1)
      WRITE (OUT,80) I,(ANS(N),N=START,100)
      IF (I.EQ.1) GO TO 40
      CALL MPGCDA (TEMP,LAST,TEMP)
      CALL MPOUT (TEMP,ANS,100,-1)
      WRITE (OUT,90) (ANS(N),N=START,100)
40    CALL MPSTR (TEMP,LAST)
50    CONTINUE
      RETURN
```

```
C
C
60    FORMAT (12H DETERMINANT,I3,2H =,I8)
70    FORMAT (5X,3HGCD,8X,1H=,I8)
80    FORMAT (12H DETERMINANT,I3,2H =,100A1)
90    FORMAT (5X,3HGCD,8X,1H=,100A1)
      END
      SUBROUTINE DIV(A,B,GCD,INV)
C
C  DIV FINDS THE GREATEST COMMON DIVISOR OF 2 NUMBERS A AND B.
C   THE GCD IS RETURNED IN GCD AND INV*A + N*B = GCD FOR SOME N.
C
      IMPLICIT INTEGER (A-Z)
      IF(A.NE.0)GO TO 5
      GCD=B
      INV=1
      RETURN
5     S=1
      GCD=A
      INV=0
      NUM=B
C
10    RATIO=NUM/GCD
      TEMP=MOD(INV-S*RATIO,B)
      INV=S
      S=TEMP
      IF(INV.LT.0)INV=INV+B
      TEMP=NUM-RATIO*GCD
      IF(TEMP.EQ.0)RETURN
      NUM=GCD
      GCD=TEMP
      GO TO 10
      END
      SUBROUTINE FDRANK (A,NROW,NCOL,PRIME)
C
C  THIS ROUTINE FINDS THE RANK OF MATRIX A MODULO PRIME.
C  THE MATRIX IS READ IN A ROW AT A TIME FROM UNIT INMAT OR INBIN.
C  EACH ROW IS IMMEDIATELY COPIED. THE NEW ROW, FOR EXAMPLE J,
C  IS CHECKED FOR LINEAR DEPENDENCE ON THE EARLIER ROWS.
C  IF DEPENDENT, ROW J IS OVERWRITTEN. IF INDEPENDENT, A(NROW,J) IS SET
C  TO A NON ZERO COLUMN OF ROW J, AND THE COPIED JTH ROW
C  IS WRITTEN OUT ONTO DETMAT.
C
C  IF THE MATRIX RANK IS LESS THAN THE NUMBER OF COLUMNS,NOTFCR IS
C  SET AND NUMDET INDEPENDENT LAST COLUMNS ARE LOOKED FOR.
C  OTHERWISE, NUMDET INDEPENDENT LAST ROWS ARE SOUGHT.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(NROW,NCOL),FMT(3),DETS(10)
      LOGICAL EARLY,NOTFCR
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT(3)
      COMMON /DET/ DETS,NUMDET,NACTRO,NACTCL,DETMAT,EARLY,NOTFCR
C
C  READ IN A ROW, COPY IT AND REDUCE IT MOD PRIME.
C
```

```
      ROW=1
      DO 90 K=1,NACTRO
      IF(INMAT.GT.0)READ(INMAT,FMT)(A(ROW,J),J=1,NCOL)
      IF(INMAT.LT.0)READ(INBIN)(A(ROW,J),J=1,NCOL)
      DO 20 J=1,NCOL
      A(ROW+1,J)=A(ROW,J)
      A(ROW,J)=MOD(A(ROW,J),PRIME)
20    IF(A(ROW,J).LT.0)A(ROW,J)=A(ROW,J)+PRIME
      IF(ROW.EQ.1)GO TO 50
      IF(NUMCOL.EQ.1)GO TO 41
C
C  CHECK NEW ROW FOR LINEAR DEPENDENCE.
C
      IEND=MINO(ROW-1,NCOL-1)
      DO 40 I=1,IEND
      COL=A(NROW,I)
      IF(A(ROW,COL).EQ.0)GO TO 40
      KILFAC=A(ROW,COL)
      DO 30 J=1,NCOL
30    A(ROW,J)=MOD(A(ROW,J)+A(I,J)*KILFAC,PRIME)
40    CONTINUE
      IF(ROW.GT.NCOL)GO TO 41
50    DO 60 J=1,NCOL
      IF(A(ROW,J).NE.0)GO TO 70
60    CONTINUE
      GO TO 90
C
C  LOOK FOR A NEW LAST ROW
C
41    COL=A(NROW,NCOL)
      IF(A(ROW,COL).EQ.0)GO TO 90
      ROWEND=ROW-1
      DO 45 I=NCOL,ROWEND
      IF(A(ROW,COL).EQ.A(I,COL))GO TO 90
45    CONTINUE
48    IF(ROW.LT.NCOL+NUMDET-1)GO TO 88
      GO TO 97
C
C  NEW ROW IS LINEARLY INDEPENDENT.
C
70    A(NROW,ROW)=J
      IF(ROW.EQ.NCOL)GO TO 48
80    CALL DIV(A(ROW,J),PRIME,DUMMY,INV)
      DO 85 J=1,NCOL
85    A(ROW,J)=MOD(A(ROW,J)*(PRIME-INV),PRIME)
88    WRITE(DETMAT)(A(ROW+1,J),J=1,NCOL)
      ROW=ROW+1
90    CONTINUE
      NACTRO=ROW-1
      IF(NACTRO.LT.NCOL)GO TO 100
      NUMDET=ROW-NCOL
95    WRITE(DETMAT)(A(NROW,J),J=1,NCOL)
      RETURN
97    NACTRO=NCOL+NUMDET-1
      WRITE(DETMAT)(A(ROW+1,J),J=1,NCOL)
```

```
      GO TO 95
C
C  LOOK FOR LINEARLY INDEPENDENT LAST COLUMNS.
C
100   NOTFCR=.TRUE.
      NONZER=0
      DO 190 J=1,NCOL
      IF(A(NACTRO,J).EQ.0)GO TO 190
      A(NROW,NACTRO+NONZER)=J
      NONZER=NONZER+1
      IF(NONZER.EQ.NUMDET)GO TO 110
190   CONTINUE
      NUMDET=NONZER
110   NACTCL=NACTRO+NONZER-1
      WRITE(DETMAT)(A(NROW,J),J=1,NACTCL)
      RETURN
      END
      SUBROUTINE GFPDET (A,NROW,NCOL,PRIME)
C
C  GFPDET CALCULATES NUMDET DETERMINANTS OF SQUARE SUBMATRICES OF
C  ARRAY A OF FULL RANK. THE DETERMINANTS ARE CALCULATED MODULO
C  THE PRIME PRIME AND STORED IN DETS. FURTHER EXPLANATIONS CAN BE
C  FOUND IN DETCAL.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(NROW,NCOL),DETS(10),FMT(3)
      LOGICAL EARLY,NOTFCR
      COMMON /DET/ DETS,NUMDET,NACTRO,NACTCL,DETMAT,EARLY,NOTFCR
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT(3)
C
C  READ IN MATRIX
C
      REWIND DETMAT
      DO 5 I=1,NACTRO
      READ(DETMAT)(A(I,J),J=1,NCOL)
      DO 5 J=1,NCOL
      A(I,J)=MOD(A(I,J),PRIME)
5     IF(A(I,J).LT.0)A(I,J)=A(I,J)+PRIME
      READ(DETMAT)(A(NACTRO+1,J),J=1,NACTCL)
C
      IF(NACTRO.EQ.1)GO TO 200
      IF(NCOL.EQ.1)GO TO 250
      DETERM=1
      CASES=NCOL-1
      IF (NOTFCR) CASES=NACTRO-1
10    DO 70 ROW=1,CASES
      COL=A(NACTRO+1,ROW)
      IF(A(ROW,COL).NE.0)GO TO 20
C
C  BAD CASE FOR THIS LIST OF PRIMES
C
      NACTRO=0
      RETURN
C
C  KILL ENTRIES IN COL COL
```

```
C
20    DETERM=MOD(DETERM*A(ROW,COL),PRIME)
      CALL DIV (A(ROW,COL),PRIME,DUMMY,INV)
      DO 40 K=1,NACTCL
      J=A(NACTRO+1,K)
      IF(J.EQ.COL)GO TO 40
      IF (A(ROW,J).EQ.0) GO TO 40
      KILFAC=MOD(A(ROW,J)*INV,PRIME)
      ISTART=ROW+1
      DO 30 I=ISTART,NACTRO
30    A(I,J)=MOD(A(I,J)+(PRIME-A(I,COL))*KILFAC,PRIME)
40    CONTINUE
70    CONTINUE
C
C  SWAP LAST ROW (OR COLUMN)
C
      IF (NOTFCR) GO TO 90
      COL=A(NACTRO+1,NCOL)
78    DO 80 K=1,NUMDET
80    DETS(K)=MOD(DETERM*A(CASES+K,COL),PRIME)
      RETURN
90    DO 100 K=1,NUMDET
      COL=A(NACTRO+1,CASES+K)
      DETS(K)=MOD(DETERM*A(NACTRO,COL),PRIME)
100   CONTINUE
110   RETURN
C
C  TRIVIAL CASES
C
200   DO 210 K=1,NUMDET
210   DETS(K)=A(1,K)
      RETURN
250   DO 260 K=1,NACTRO
260   DETS(K)=A(K,1)
      RETURN
      END
      SUBROUTINE MATOUT(A,M,N)
C
C THIS SUBROUTINE PRINTS OUT A MATRIX
C
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(N),FMT(3)
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT(3)
      DO 10 I =1,M
      IF(INMAT.GT.0)READ(INMAT,FMT)(A(J),J=1,N)
      IF(INMAT.LT.0)READ(INBIN)(A(J),J=1,N)
10    WRITE(OUT,FMT)(A(J),J=1,N)
      RETURN
      END
      SUBROUTINE MODIAG(A,NROW,NCOL,MODULO,OPT)
C
C  THIS ROUTINE DIAGONALISES AN INTEGER MATRIX MODULO A GIVEN
C   PRIME POWER MODULO. THE ALGORITHM USED IS DISCUSSED IN SECTION 8
C   OF THE EUROSAM PAPER.
C   VARIABLE NAMES ARE THE SAME AS IN IMDIAG WHERE POSSIBLE,
```

```
C   AND EXPLANATIONS CAN BE FOUND THERE.
C   DIFFERENT VARIABLE NAMES ARE
C  GCD - GREATEST COMMON DIVISOR OF A MATRIX ENTRY AND MODULO.
C  MINGCD - MINIMUM GCD FOUND SO FAR.
C  INV - INVERSE OF A MATRIX ENTRY MODULO MODULO.
C  MININV - INVERSE ASSOCIATED WITH MINGCD.
C  GCD AND INV ARE CALCULATED BY SUBROUTINE DIV WHICH PERFORMS THE
C   FORWARD EXTENDED EUCLIDEAN ALGORITHM.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(NROW,NCOL),FMT(3)
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT(3)
      COMMON /MP/ B,T,M,LUN,MXR,WSPACE(500)
C
C  READ IN MATRIX
C
      NREL=NROW-1
      IF(INMAT.GT.0)GO TO 8
      DO 5 I=1,NREL
5     READ(INBIN)(A(I,J),J=1,NCOL)
      IF(OPT.EQ.0)GO TO 12
      READ(INBIN)(A(NROW,J),J=1,NCOL)
      GO TO 14
8     DO 10 I=1,NREL
10    READ(INMAT,FMT)(A(I,J),J=1,NCOL)
      IF(OPT.EQ.0)GO TO 12
      READ(INMAT,FMT)(A(NROW,J),J=1,NCOL)
      GO TO 14
12    DO 13 J=1,NCOL
13    A(NROW,J)=J
14    DO 15 I=1,NREL
      DO 15 J=1,NCOL
      A(I,J)=MOD(A(I,J),MODULO)
15    IF(A(I,J).LT.0)A(I,J)=A(I,J)+MODULO
      CASES=MINO(NREL,NCOL)
C
      DO 150 ITER=1,CASES
      MINGCD=MODULO
C
C  SEARCH FOR MINIMUM GCD. IF MINGCD=1 GO STRAIGHT TO KILLING OFF.
C
      DO 20 I=ITER,NREL
      DO 20 J=ITER,NCOL
      CALL DIV(A(I,J),MODULO,GCD,INV)
      IF(GCD.GE.MINGCD)GO TO 20
      MINGCD=GCD
      MINI=I
      MINJ=J
      MININV=INV
      IF(MINGCD.EQ.1)GO TO 50
20    CONTINUE
C
C  NO RELATIVELY PRIME ELEMENT IN MATRIX. TEST FOR ALL ZEROS.
C
      IF(MINGCD.EQ.MODULO)GO TO 160
```

```
C
C  MULTIPLY ROW TO MAKE KILLING ELEMENT A POWER OF A PRIME
C
50    DO 30 J=ITER,NCOL
30    A(MINI,J)=MOD(A(MINI,J)*MININV,MODULO)
C
C  SWAP KILLING ELEMENT TO TOP LEFTHAND CORNER OF SUBMATRIX.
C  KILL OFF ELEMENTS IN THE LEFTHAND COLUMN.
C
      IF(MINI.NE.ITER)CALL ROWSWP(A,NROW,NCOL,ITER,MINI)
      IF(MINJ.NE.ITER)CALL COLSWP(A,NROW,NCOL,ITER,MINJ)
      IF(ITER.EQ.CASES)GO TO 110
      START=ITER+1
      DO 100 I=START,NREL
      IF(A(I,ITER).EQ.0)GO TO 100
      KILFAC=A(I,ITER)/A(ITER,ITER)
      DO 80 J=START,NCOL
80    A(I,J)=MOD(A(I,J)+(MODULO-A(ITER,J))*KILFAC,MODULO)
100   CONTINUE
110   A(ITER,ITER)=MINGCD
      IF(OUTLEV.LT.3)GO TO 150
      DIAGEL=-A(ITER,ITER)
      LENGTH=0
      IF(ITER.EQ.CASES)GO TO 140
      DO 120 J=START,NCOL
      IF(A(ITER,J).EQ.0)GO TO 120
      LENGTH=LENGTH+2
      WSPACE(LENGTH)=A(NROW,J)
      WSPACE(LENGTH-1)=A(ITER,J)
120   CONTINUE
      IF(LENGTH.EQ.0)GO TO 140
      WRITE(OUT,200) DIAGEL,A(NROW,ITER),(WSPACE(I),I=1,LENGTH)
      GO TO 150
140   WRITE(OUT,210) DIAGEL,A(NROW,ITER)
150   CONTINUE
C
C  OUTPUT STRUCTURE OF DIAGONALISED MATRIX.
C
160   NONTRV=0
      DO 180 K=1,CASES
      IF(A(K,K)-1)190,180,170
170   NONTRV=NONTRV+1
      WSPACE(NONTRV)=A(K,K)
180   CONTINUE
      K=CASES+1
190   RANK=K-1
      WRITE(OUT,230)NREL,NCOL,RANK
      IF(NONTRV.EQ.0)WRITE(OUT,240)
      IF(NONTRV.GT.0)WRITE(OUT,250)(WSPACE(I),I=1,NONTRV)
      RETURN
C
200   FORMAT(I7,2H *,I5,3H = ,8(I6,2H *,I5)/(17X,8(I6,2H *,I5)))
210   FORMAT(I7,2H *,I5,2H =,6X,1H0)
230   FORMAT(17HONUMBER OF ROWS =,I12/20H NUMBER OF COLUMNS =,I9/
     1        7HORANK =,I10)
```

```
240   FORMAT(42HOTHERE ARE NO NONTRIVIAL DIAGONAL ELEMENTS)
250   FORMAT(37HOTHE NONTRIVIAL DIAGONAL ELEMENTS ARE,15I6/(37X,15I6))
      END
      SUBROUTINE REDROW (A,NROW,NCOL,TOP,MAXCOL)
C
C  THIS SUBROUTINE ATTEMPTS TO LESSEN THE RISK OF COEFFICIENT EXPLOSION
C   IN THE PARTIALLY PROCESSED MATRIX A, BY REDUCING THE MATRIX ENTRIES.
C   THE BASIC STRATEGY IS TO SUBTRACT ROWS FROM EACH OTHER UNTIL NO MORE
C   IMPROVEMENT IS POSSIBLE, WHERE GOODNESS OF A ROW IS DETERMINED BY
C   THE SUM OF THE ABSOLUTE VALUES OF ITS ELEMENTS.
C   THE ROWS ARE SORTED INTO DESCENDING ORDER OF THE SIZE OF THE ELEMENT
C   IN COLUMN MAXCOL OF EACH ROW, AND THE ORDER IS POINTED TO BY THE
C   VECTOR ORDER IN MP.
C   NOTE THAT ALL ROWS HAVE POSITIVE MAXCOL ELEMENTS (WE MULTIPLY BY -1
C   IF NECESSARY).
C   N.B. THE MAXCOL ELEMENT MUST IMPROVE AS WELL AS THE ROWSUM FOR THE
C        SUBTRACTION TO BE PERFORMED.
C
C  VARIABLES USED ARE
C  NROW,NCOL - AS IN IMDIAG.
C  TOP,BOTTOM - TOP AND BOTTOM ROWS RESPECTIVELY OF THE SUBMATRIX TO BE
C               ECHELONIZED.
C  SECOND,PENUL - 2ND AND 2ND LAST ROWS RESPECTIVELY OF THE SUBMATRIX.
C  ROW - DO LOOP VARIABLE INDICATING THE ROW WE ARE TRYING TO IMPROVE.
C  SUB - ANOTHER DO LOOP VARIABLE GIVING THE ROW WE SUBTRACT FROM ROW.
C  ORDROW,ORDSUB - THE ACTUAL ROWS POINTED TO BY ROW AND SUB IN ORDER.
C  ROWSUM - SUM OF ABSOLUTE VALUES OF ENTRIES IN ROW.
C  NEWSUM - SUM OF ABSOLUTE VALUES OF ENTRIES IN ROW - SUB.
C  POS,ORDNEX - VARIABLES HELPING TO PLACE A NEWLY FORMED ROW IN ITS
C               CORRECT POSITION.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(NROW,NCOL)
      COMMON /MP/ B,T,MMP,LUN,MXR,ORDER(500)
      BOTTOM=NROW-1
22    IF (TOP.GE.BOTTOM) RETURN
C
C  SET UP POINTER VECTOR
C
      DO 5 I=TOP,BOTTOM
5     ORDER(I)=I
C
C  STANDARDISE ROWS
C
      DO 15 I=TOP,BOTTOM
      IF (A(I,MAXCOL).GE.0) GO TO 15
      DO 10 J=TOP,NCOL
10    A(I,J)=-A(I,J)
15    CONTINUE
C
C  SORT ROWS INTO DESCENDING ORDER OF MAXCOL ELEMENTS
C
      SECOND=TOP+1
      DO 35 I=SECOND,BOTTOM
      J=I-1
```

```
20    K=ORDER(J)
      L=ORDER(J+1)
      IF(A(K,MAXCOL).GE.A(L,MAXCOL))GO TO 35
30    ORDER(J+1)=K
      ORDER(J)=L
      J=J-1
      IF (J.GE.TOP) GO TO 20
35    CONTINUE
C
C  GO THROUGH MATRIX SUBTRACTING PAIRS OF ROWS,
C   TESTING FOR IMPROVEMENT.
C
      PENUL=BOTTOM-1
      DO 85 ROW=TOP,PENUL
      NEXROW=ROW+1
40    ORDROW=ORDER(ROW)
      SIGN=1
C
C  CALCULATE ROW SUM FOR ROW ROW
C
      ROWSUM=0
      DO 45 J=TOP,NCOL
45    ROWSUM=ROWSUM+IABS(A(ORDROW,J))
      DO 85 SUB=NEXROW,BOTTOM
      ORDSUB=ORDER(SUB)
      IF(A(ORDSUB,MAXCOL).EQ.0)GO TO 48
C
C  TRY A MULTIPLE SUBTRACTION.
C
      MULT=A(ORDROW,MAXCOL)/A(ORDSUB,MAXCOL)
      IF(A(ORDROW,MAXCOL)-MULT*A(ORDSUB,MAXCOL).LE.A(ORDSUB,MAXCOL)/2)
     1    GO TO 46
      MULT=MULT+1
      SIGN=-1
46    NEWSUM=0
      DO 47 J=TOP,NCOL
47    NEWSUM=NEWSUM+IABS(A(ORDROW,J)-MULT*A(ORDSUB,J))
      IF(NEWSUM.LT.ROWSUM)GO TO 52
      IF(MULT.EQ.1)GO TO 85
C
C  TRY A SINGLE SUBTRACTION.
C
48    NEWSUM=0
      DO 50 J=TOP,NCOL
50    NEWSUM=NEWSUM+IABS(A(ORDROW,J)-A(ORDSUB,J))
      IF (NEWSUM.GE.ROWSUM) GO TO 85
      MULT=1
      SIGN=1
C
C  SUBTRACTED ROW IS BETTER THAN OLD ONE.
C  REPLACE TOP ROW AND FIND ITS CORRECT POSITION
C
52    DO 55 J=TOP,NCOL
55    A(ORDROW,J)=(A(ORDROW,J)-MULT*A(ORDSUB,J))*SIGN
      POS=ROW
```

```
60    ORDNEX=ORDER(POS+1)
      IF(A(ORDROW,MAXCOL).GE.A(ORDNEX,MAXCOL))GO TO 40
C
C  SWAP 2 ROWS.
C
80    ORDER(POS)=ORDNEX
      ORDER(POS+1)=ORDROW
      POS=POS+1
      IF (POS.LT.BOTTOM) GO TO 60
      GO TO 40
85    CONTINUE
      RETURN
      END
      SUBROUTINE ROWSWP(A,M,N,K,L)
C
C  ROWSWP SWAPS THE KTH AND LTH ROWS OF ARRAY A
C  AS IN COLSWP, K IS LESS THAN L, AND WE NEED ONLY SWAP
C  FROM THE KTH COLUMN ONWARDS.
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION A(M,N)
      DO 100 J=K,N
      TEMP=A(K,J)
      A(K,J)=A(L,J)
100   A(L,J)=TEMP
      RETURN
      END
      SUBROUTINE SMITH(A,NROW,NCOL)
C
C  THIS ROUTINE MANIPULATES THE DIAGONAL ELEMENTS OF THE MATRIX A,
C  SO THAT THEY SATISFY THE DIVISIBILITY CONDITIONS OF THE SMITH
C  NORMAL FORM. CALCULATIONS OF GREATEST COMMON DIVISORS AND
C  LOWEST COMMON MULTIPLES ARE PERFORMED USING DIV.
C
      IMPLICIT INTEGER(A-Z)
      DIMENSION A(NROW,NCOL)
      CASES=MINO(NROW-1,NCOL)
C
      DO 5 I=1,CASES
5     A(I,I)=IABS(A(I,I))
      IF(CASES.LE.1)RETURN
      I=1
10    IF(A(I,I)-1)40,20,30
20    I=I+1
      IF(I.LT.CASES)GO TO 10
      RETURN
30    J=I+1
      IF(MOD(A(J,J),A(I,I)).EQ.0)GO TO 20
      BIG=MAXO(A(I,I),A(J,J))
      LIT=MINO(A(I,I),A(J,J))
      CALL DIV(BIG,LIT,GCD,DUMMY)
      A(J,J)=A(J,J)*A(I,I)/GCD
      A(I,I)=GCD
      I=MAXO(I-1,1)
      GO TO 10
```

```
40    RETURN
      END
      SUBROUTINE RUNIMD(A,NROW,NCOL,OPT)
C
C  THIS IS AN OPTION SUBROUTINE FOR DIAGONALISATION
C
      IMPLICIT INTEGER (A-Z)
      LOGICAL NORED
      DIMENSION A(NROW,NCOL),FMT(3)
      COMMON /FORM/ FMT,IN,OUT,INMAT,INBIN,OUTLEV,NBITS,IDENT(3)
      COMMON /IMDPAR/ LITROW,BOUND,POWER2,INCR,MAXINT
      DATA OPTEL,OPTRR,OPTRC,OPTWM,OPTGM/2HEL,2HRR,2HRC,2HWM,2HGM/
      DATA OPTOL,OPTPE,OPTCP,OPTSM,OPTEX/2HOL,2HPE,2HCP,2HSM,2HEX/
      NREL=NROW-1
      MAXINT=2*(2**(NBITS-2)-1)+1
      TOP=1
      CASES=MINO(NREL,NCOL)
      IF(INMAT.LT.0)GO TO 15
      DO 10 I=1,NREL
10    READ(INMAT,FMT)(A(I,J),J=1,NCOL)
      IF(OPT.GE.0)GO TO 20
      READ(INMAT,FMT)(A(NROW,J),J=1,NCOL)
      GO TO 40
15    DO 18 I=1,NREL
18    READ(INBIN)(A(I,J),J=1,NCOL)
      IF(OPT.GE.0)GO TO 20
      READ(INBIN)(A(NROW,J),J=1,NCOL)
      GO TO 40
20    DO 30 J=1,NCOL
30    A(NROW,J)=J
      IF(OPT.NE.0)GO TO 40
      CALL IMDIAG(A,NROW,NCOL,TOP,CASES,.FALSE.)
      RETURN
40    WRITE(OUT,310)
      READ(IN,210)IMDOPT,PARAM
      IF(IMDOPT.EQ.OPTEL)GO TO 50
      IF(IMDOPT.EQ.OPTRR)GO TO 60
      IF(IMDOPT.EQ.OPTRC)GO TO 70
      IF(IMDOPT.EQ.OPTWM)GO TO 80
      IF(IMDOPT.EQ.OPTGM)GO TO 100
      IF(IMDOPT.EQ.OPTOL)GO TO 120
      IF(IMDOPT.EQ.OPTPE)GO TO 130
      IF(IMDOPT.EQ.OPTCP)GO TO 150
      IF(IMDOPT.EQ.OPTSM)GO TO 160
      IF(IMDOPT.EQ.OPTEX)RETURN
      WRITE(OUT,320)
      GO TO 40
C
C  ELIMINATIONS
C
50    NORED=.FALSE.
      IF(PARAM.EQ.-1)NORED=.TRUE.
      IF(PARAM.LE.0)PARAM=CASES+1-TOP
      CALL IMDIAG(A,NROW,NCOL,TOP,TOP+PARAM-1,NORED)
      IF(NORED)GO TO 40
```

```
      TOP=MINO(TOP+PARAM,CASES)
      GO TO 40
C
C  ROW REDUCTIONS
C
60    IF(PARAM.GT.0)GO TO 65
      MAX=0
      MAXCOL=TOP
      DO 62 I=TOP,NREL
      DO 62 J=TOP,NCOL
      IF(IABS(A(I,J)).LE.MAX)GO TO 62
      MAX=IABS(A(I,J))
      MAXCOL=J
62    CONTINUE
      PARAM=MAXCOL-TOP+1
65    CALL REDROW(A,NROW,NCOL,TOP,TOP+PARAM-1)
      GO TO 40
C
C  COLUMN REDUCTIONS
C
70    IF(PARAM.GT.0)GO TO 75
      MAX=0
      MAXROW=TOP
      DO 72 I=TOP,NREL
      DO 72 J=TOP,NCOL
      IF(IABS(A(I,J)).LE.MAX)GO TO 72
      MAX=IABS(A(I,J))
      MAXROW=I
72    CONTINUE
      PARAM=MAXROW-TOP+1
75    CALL REDCOL(A,NROW,NCOL,TOP,TOP+PARAM-1)
      GO TO 40
C
C  SAVE (WRITE) MATRIX
C
80    IMDREL=NROW-TOP
      IMDCOL=NCOL+1-TOP
      IF(PARAM.EQ.0)PARAM=20
      IF(PARAM.LT.0)GO TO 95
      REWIND PARAM
      WRITE(PARAM,340)IMDREL,IMDCOL,IDENT,FMT
      DO 90 I=TOP,NROW
90    WRITE(PARAM,FMT)(A(I,J),J=TOP,NCOL)
      GO TO 40
95    PARBIN=-PARAM
      REWIND PARBIN
      WRITE(PARBIN,340)IMDREL,IMDCOL,IDENT
      DO 98 I=TOP,NROW
98    WRITE(PARBIN)(A(I,J),J=TOP,NCOL)
C
C  RESTORE (GET) MATRIX
C
100   IF(PARAM.EQ.0)PARAM=20
      IF(PARAM.LT.0)GO TO 115
      REWIND PARAM
```

```
      READ(PARAM,340)IMDREL,IMDCOL,IDENT,FMT
      TOP=NROW-IMDREL
      DO 110 I=TOP,NROW
110   READ(PARAM,FMT)(A(I,J),J=TOP,NCOL)
      GO TO 40
115   PARBIN=-PARAM
      REWIND PARBIN
      READ(PARBIN,340)IMDREL
      TOP=NROW-IMDREL
      DO 118 I=TOP,NROW
118   READ(PARBIN)(A(I,J),J=TOP,NCOL)
      GO TO 40
C
C  CHANGE THE OUTPUT LEVEL
C
120   OUTLEV=IABS(PARAM)
      GO TO 40
C
C  PRINT ROW,DIAGONAL, OR COLUMN.
C
130   IF(PARAM.LT.0)WRITE(OUT,FMT)(A(-PARAM,J),J=TOP,NCOL)
      IF(PARAM.EQ.0)WRITE(OUT,FMT)(A(I,I),I=1,TOP)
      IF(PARAM.GT.0)WRITE(OUT,FMT)(A(I,PARAM),I=TOP,NREL)
      GO TO 40
C
C  CHANGE THE INPUT FORMAT
C
140   WRITE(OUT,350)
      READ(IN,220)FMT
      GO TO 40
C
C  CHANGE PARAMETERS OF IMDPAR
C
150   IF(PARAM.LT.0)GO TO 140
      WRITE(OUT,360)
      READ(IN,230)NEWLIT,NEWBOU,NEWPOW,NEWINC
      IF(NEWLIT.NE.0)LITROW=NEWLIT
      IF(LITROW.LT.0)LITROW=MAXINT
      IF(NEWPOW.NE.0)POWER2=IABS(NEWPOW)
      IF(NEWBOU.NE.0)BOUND=MAXO(POWER2+1,NEWBOU)
      IF(NEWINC.NE.0)INCR=MAXO(0,NEWINC)
      GO TO 40
C
C  CONVERT DIAGONAL FORM TO SMITH NORMAL FORM
C
160   CALL SMITH(A,NROW,NCOL)
      GO TO 40
C
210   FORMAT(A2,I3)
220   FORMAT(3A4)
230   FORMAT(4I3)
310   FORMAT(18H IMD OPTION(A2,I3))
320   FORMAT(19H ILLEGAL IMD OPTION)
340   FORMAT(2I4,6A4)
350   FORMAT(15H INPUT FMT(3A4))
```

```
360    FORMAT(36H INPUT LITROW,BOUND,POWER2,INCR(4I3))
       END
       SUBROUTINE REDCOL(A,NROW,NCOL,TOP,MAXROW)
C
C  COLUMN REDUCTION ROUTINE. SAME ALGORITHM AS FOR ROW REDUCTIONS.
C
       IMPLICIT INTEGER (A-Z)
       DIMENSION A(NROW,NCOL)
       COMMON /MP/B,T,MMP,LUN,MXR,ORDER(500)
       BOTTOM=NROW-1
       IF(TOP.GE.NCOL)RETURN
C
       DO 5 J=TOP,NCOL
5      ORDER(J)=J
C
       DO 15 J=TOP,NCOL
       IF(A(MAXROW,J).GE.0)GO TO 15
       DO 10 I=TOP,BOTTOM
10     A(I,J)=-A(I,J)
15     CONTINUE
C
       SECOND=TOP+1
       DO 35 I=SECOND,NCOL
       J=I-1
20     K=ORDER(J)
       L=ORDER(J+1)
       IF(A(MAXROW,K).GE.A(MAXROW,L))GO TO 35
30     ORDER(J+1)=K
       ORDER(J)=L
       J=J-1
       IF(J.GE.TOP)GO TO 20
35     CONTINUE
C
       PENUL=NCOL-1
       DO 85 COL=TOP,PENUL
       NEXCOL=COL+1
40     ORDCOL=ORDER(COL)
       SIGN=1
       COLSUM=0
       DO 45 I=TOP,BOTTOM
45     COLSUM=COLSUM+IABS(A(I,ORDCOL))
       DO 85 SUB=NEXCOL,NCOL
       ORDSUB=ORDER(SUB)
       IF(A(MAXROW,ORDSUB).EQ.0)GO TO 48
       MULT=A(MAXROW,ORDCOL)/A(MAXROW,ORDSUB)
       IF(A(MAXROW,ORDCOL)-MULT*A(MAXROW,ORDSUB).LE.A(MAXROW,ORDSUB)/2)
     1       GO TO 46
       MULT=MULT+1
       SIGN=-1
46     NEWSUM=0
       DO 47 I=TOP,BOTTOM
47     NEWSUM=NEWSUM+IABS(A(I,ORDCOL)-A(I,ORDSUB)*MULT)
       IF(NEWSUM.LT.COLSUM)GO TO 52
       IF(MULT.EQ.1)GO TO 85
48     NEWSUM=0
```

```
      DO 50 I=TOP,BOTTOM
50    NEWSUM=NEWSUM+IABS(A(I,ORDCOL)-A(I,ORDSUB))
      IF(NEWSUM.GE.COLSUM)GO TO 85
      MULT=1
      SIGN=1
C
52    DO 55 I=TOP,BOTTOM
55    A(I,ORDCOL)=(A(I,ORDCOL)-MULT*A(I,ORDSUB))*SIGN
      POS=COL
60    ORDNEX=ORDER(POS+1)
      IF(A(MAXROW,ORDCOL).GE.A(MAXROW,ORDNEX))GO TO 40
C
80    ORDER(POS)=ORDNEX
      ORDER(POS+1)=ORDCOL
      POS=POS+1
      IF(POS.LT.NCOL)GO TO 60
      GO TO 40
85    CONTINUE
      RETURN
      END
```

INSTRUCTIONS FOR IMD PROGRAM - Integer Matrix Diagonalisation

IMD is a computer program which computes a diagonal form for an integer matrix. A description of the program and some of its applications appears in G.Havas and L.S.Sterling 'Integer Matrices and Abelian Groups', Proc. 1979 EUROSAM Conference. The program is written in a superset of 1966 ANSI standard FORTRAN and is very portable. Notation is used according to FORTRAN.

The program requires a matrix as input. The default unit for the matrix is 10 (to change this see the CI command). The first line of the matrix input unit must contain the number of rows, the number of columns, both in I4 format, an identifier for the matrix, and the input format for the matrix, both in 3A4 format. The matrix then follows row by row according to the input format specified in the first input line. (Optionally, the matrix can be in binary as described below.)

The default units are:- 5 for the input unit IN, 6 for the output unit OUT, and 12 for DETMAT, a temporary unit for reading and writing the matrix in binary during determinant calculations. IN,OUT,DETMAT can be changed in the block data routine.

When the program is loaded, it will identify itself, then prompt OPTION(A2,I3) to indicate its readiness to accept a command. As indicated, the command should be entered as an A2 field, optionally followed by an integer parameter in I3 format.

COMMANDS

ID n

This command calls subroutine RUNID which controls the integer diagonalisation routine, IMDIAG. If n.eq.0, diagonalisation occurs automatically using default parameters. If n.ne.0, the user must specify ID option commands as described below. If n.ge.0, the columns are given the labels 1 - m, where m is the no. of columns. These labels are used in some output messages. If the user wishes to use his own labels, where a label is an integer compatible with the input format, he does an ID n with n.lt.0. The labels are then the last line of input.

MD n

This command performs diagonalisation modulo n (if n.gt.0). As in ID, the columns are given the labels 1 - m, if n.gt.0. Otherwise user labels are required as the last line of input. If n.le.0, the program prompts you to specify the number modulo which you want to diagonalise. Two points should be stressed. Firstly, to guarantee correct results, a modulus less than the square root of the largest machine-representable integer should be used. Otherwise overflow may occur. Secondly, the modulus MUST be a power of a prime.

DC n

This command first computes the rank, r say, of the matrix. Then up to abs(n) rxr determinants of different submatrices are calculated, and their g.c.d. computed. The submatrices are different in the last row (or last column if r is less than the no. of columns). If n.gt.0, a Hadamard bound is computed to guarantee the correctness of the determinant. A faster test, practically but not theoretically correct, is used when n.lt.0. See the paper for details. The default value for n is 1.

PM

Print out the matrix.

OL n

The level of output produced by the program is controlled by the variable OUTLEV, which may be set to 1 (least output), 2 (intermediate output), 3 (maximum output) and 4 (diagnostic output). OL n alters OUTLEV to abs(n). The program may be run in either interactive mode or batch mode. The only difference is that the latter echoes commands. The initial mode is interactive. Batch mode may be entered by an OL n command with n.lt.0, and interactive mode restored by OL n with n.gt.0.

CI n

This command changes the matrix input unit to abs(n). If n.gt.0, the matrix is read in according to the format specified in the top row. If n.lt.0, the matrix is read in binary. Beware of using unit 12, i.e. unit DETMAT.

GP n

This generates optimal primes used in calculating determinants for a machine with an integer wordsize of n (n.ge.18). If OL.eq.17 a copy of the primes and moduli is written on unit OUT. Beware of using small primes for two reasons. Firstly the fast determinant test may be invalid. Secondly the list of primes may be unsuitable because columns where non-zero elements occur as calculated in FDRANK may be wrong. In this case, the following message appears:

THIS LIST OF PRIMES IS UNSUITABLE FOR THIS MATRIX

EX

Exit from the program.

ID option commands

An ID n with n.ne.0 enters the user interactively into the option routine RUNID, which controls the running of the diagonalisation routine IMDIAG. IMDIAG follows the basic algorithm described in the paper, together with the two heuristic modifications. Step numbers

refer to steps in the paper's algorithm description. The sequence of steps 2-8, i.e. choosing an element of minimal non-zero magnitude, checking divisibility conditions, and setting the entries of the leftmost column and top row to zero, is called an elimination. The second heuristic modification, i.e. subtracting rows from each other to decrease the size of entries in the matrix, is called a reduction. RUNID enables the user to control how, and in what order, eliminations and reductions are performed. Commands EL,RR,RC,CP do this. WM,GM save and restore, respectively, intermediate stages of calculation. PE,OL control the output. As eliminations occur, the submatrix being processed changes. Variable TOP indicates which submatrix is under consideration. Both eliminations and reductions occur in the submatrix from the TOPth row downwards and the TOPth column rightwards.

Initially RUNID reads in the matrix. When ready, the program prompts ID OPTION(A2,I3). The input commands have the same format as before:- an A2 field, followed by an integer in I3 format.

Commands are:-

EL n

If n.gt.0, n eliminations in the submatrix are performed. If n.eq.0, the whole submatrix is diagonalised. If n.eq.-1, eliminations are done until IMDIAG intends to perform a reduction. Instead control is returned to RUNID. Note that in each case TOP is reset appropriately.

RR n

The row reduction algorithm described in the paper is applied to the submatrix. If n.gt.0, column n is the column used as the sorting key. If not, the maximum element in the submatrix is found and its column is used as the key column. TOP is unchanged.

RC n

This applies the analogous column reduction routine. The key row is n (n.gt.0), or the row where the maximum element occurs. Again, TOP is unaltered.

WM n

This command writes out the submatrix on unit abs(n). The first line of output is the no. of rows, the no. of columns, the identifier, and the format if applicable. If n.gt.0, the matrix is written out according to the last format specified. If n.lt.0, it is written in binary.

GM n

The submatrix is read from unit abs(n), with TOP being reset accordingly. Note that the matrix on abs(n) must be a submatrix of the original matrix processed by RUNID, so that matrix dimensions

agree. Again the sign of n determines the format of the matrix.

OL n

As in the main program, but the batch option is not present.

PE n

If n.lt.0, row abs(n) is printed from the TOPth entry onwards. If n.eq.0, the diagonal is printed up to the TOPth entry. If n.gt.0, column n is printed from the TOPth entry onwards.

CP n

If n.lt.0, you are prompted to change the format used for reading and writing matrices. If n.ge.0, the program prompts ENTER LITROW,BOUND,POWER,INCR(4I3) These parameters control the heuristics as follows. The first heuristic modification involves step 2, choice of an element of minimal magnitude. Clearly any entry of 1 would be suitable. When, in step 8, entries in the leftmost column are set to zero, the size of entries in the top row will determine the size of entries in the resultant submatrix. Thus if there are several entries of absolute value 1, consideration may be taken of which row is best. The criterion used by the program is the sum of the absolute values of entries in a row, which is called rowsum. If a row with a 1 is found with rowsum less than LITROW, it is used immediately for elimination. Otherwise the program keeps looking for 1's, while remembering where the minimum rowsum has occurred. The default value for LITROW is 100. If you wish to take great care in choice of 1's, set LITROW to 5 say. An input of 0 for LITROW leaves its last value unchanged, while a negative input sets LITROW to MAXINT, the largest machine-representable integer, effectively removing this heuristic.

The second heuristic modification, i.e. reduction, is controlled by these other parameters. In steps 4,6,8 new elements are generated. When performing these steps the program checks the size of these entries. If after a step has finished, an entry whose absolute value is larger than 2**POWER has been found, a reduction is done. The default value for POWER is 10, giving an initial bound of 1024. After the reduction, POWER is increased by INCR (whose default value is 1), thus changing the bound before the next reduction. In default, the bound is doubled each time. N.B. Each time IMDIAG is called from RUNID, POWER is reset to its initial value. Once the bound before a reduction step is increased to more than the square root of the integer word size, there is a possibility of undetected integer overflow. In practice, overflow does not occur till later. BOUND controls when you wish to stop the incrementing process. Once POWER reaches BOUND the program terminates. The default for BOUND is 26. Again, an input of 0 for these 3 variables leaves them unchanged from their last set values.

SM

Compute the Smith normal form of the matrix, using the diagonal entries.

EX

Exit from RUNID.

INDEX OF NOTATION

$[x, y] = x^{-1}y^{-1}xy$

$[G, H] = \langle [g, h], g \in G, h \in H \rangle$

$G^n = \langle g^n, g \in G \rangle$

$G' = [G, G]$

$Z(G) = \langle z \in G : zg = gz, \forall g \in G \rangle$

$\Gamma_i(G)$ is the ith term in the lower central series of G

$I(G)$ (or $I_G(H)$) is the isolator of a normal subgroup H in a group G

$h(G)$ is the Hirsch number of a group G

$T(d, s)$ is the family of torsionfree nilpotent groups of class 2, with Hirsch number $d + s$, and commutator subgroup of rank s

$T(d, s)$ is the family of isolated nilpotent groups of class 2 with Hirsch number $d + \binom{d}{2} - s$, and d generators

C_α is the cyclic group of order α

$\emptyset$ is the identity of a group

$a|b$ denotes that a divides b, that is there is an element q such that $a = qb$

(a, b) denotes the greatest common divisor of a and b

$\lfloor x \rfloor$ denotes the greatest integer less than or equal to x

$\mathbb{Q}$ represents the rationals

$\mathbb{Z}$ represents the integers

$\mathbb{Z}_{(p)}$ represents the localisation of the integers at a prime p

$M(i, j)$ is the (i, j)th entry of a matrix

T^t is the transpose of T

$\det A$, or $\det(A)$, is the determinant of A

$d_i(M)$ is the ith determinantal divisor of M.

$e_i(M)$ is the ith invariant factor of M

$S(M)$ is the Smith normal form of an integer matrix M

$S_p(M)$ is the Smith normal form of a $\mathbb{Z}_{(p)}$-matrix M

$\mathrm{Pf}(M)$ is the Pfaffian of a skew-symmetric matrix M

$\mathrm{GL}(d, \mathbb{Z})$ is the group of invertible $d \times d$ integer matrices

$\mathrm{GL}_\lambda(2, \mathbb{Z})$ is the subgroup of $\mathrm{GL}(2, \mathbb{Z})$ consisting of matrices S such that $\lambda | S(2, 1)$

$(\gamma, \delta, \varepsilon)$ denotes the binary quadratic form $\gamma x^2 + \delta xy + \varepsilon y^2$

$\Delta(f)$ is the discriminant of f

$o(f)$ is the order of f

$u + v\sqrt{\Delta}$ represents the solution u, v of the Diophantine equation

$$x^2 - \Delta y^2 = N$$

□ denotes the end of a proof.

INDEX OF DEFINITIONS